Protocol Channels & Protocol Hopping Covert Channels

Steffen Wendzel
www.wendzel.de
Steffenwendzel (at) gmx (dot) net

mrmcd0x8h
06.09.2009

v. 1-0-3-final

Worum geht es?

- Covert Channel
  - Was ist das?
  - Nutzen
- Protocol Hopping Covert Channels
  - Definition, Funktionsweise
- Protocol Channels
  - Definition, Funktionsweise

Teil I

Einführung

Was ist ein Covert Channel?

- Covert Channel = verdeckter Kanal
- Zweck
  - Kommunikation soll versteckt ablaufen
  - Es geht darum, den Informations*austausch* zu verschleiern und nicht um eine Verschlüsselung!

Was ist ein Covert Channel? (2)

- Nützlich für …
  - Kontrolle von Botnetzen [LINGM02]
  - Informationsaustausch zw. Spionen [SCHN06]
  - Evtl. für Journalisten → versteckte Übertragung in überwachten Netzwerken
  - Jeden, der Daten geheim übertragen will!
- Dual-Use-Gut

[SCHN06] Schneier, B.: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C, Pearson Studium, 2. Aufl., Nachdr. 2006.
[LINGM02] Lingmann, T.: Datenverschlüsselung. Sichere Kommunikation mit Linux und BSD, C&L, 1. Aufl., 2002.

Was ist ein Covert Channel? (3)

- Wer wendet Covert Channels wirklich an?
- Der Nutzer eines CC darf nicht über dessen Existenz kommunizieren!
  - Verfahren muss geheim bleiben

    → daher keine genauen Aussagen über CC-Nutzung möglich!

Übliche Funktionsweise (1)

- Storage Channels (ändern Attribute)
  - TTL Channel in IPv4
  - HTTP Cookie
  - ...
- Timing Channel (ändern Reihenfolge und Zeit-Deltas)
  - Künstlich verzögerte Response nach Client-Request
  - N Pakete in abgewandelter Reihenfolge senden

Übliche Funktionsweise (2)

- Lokal
  - Bspw. Datei existiert (nicht)
- Nicht lokal
  - Bspw. POP3 Case-Channel (→ ReTr ...)
- Aktiv
  - Channel generiert eigenes Austauschereignis
- Passiv
  - Channel nutzt fremdes Ereignis (→ Nushu)

Unterbindung schwierig

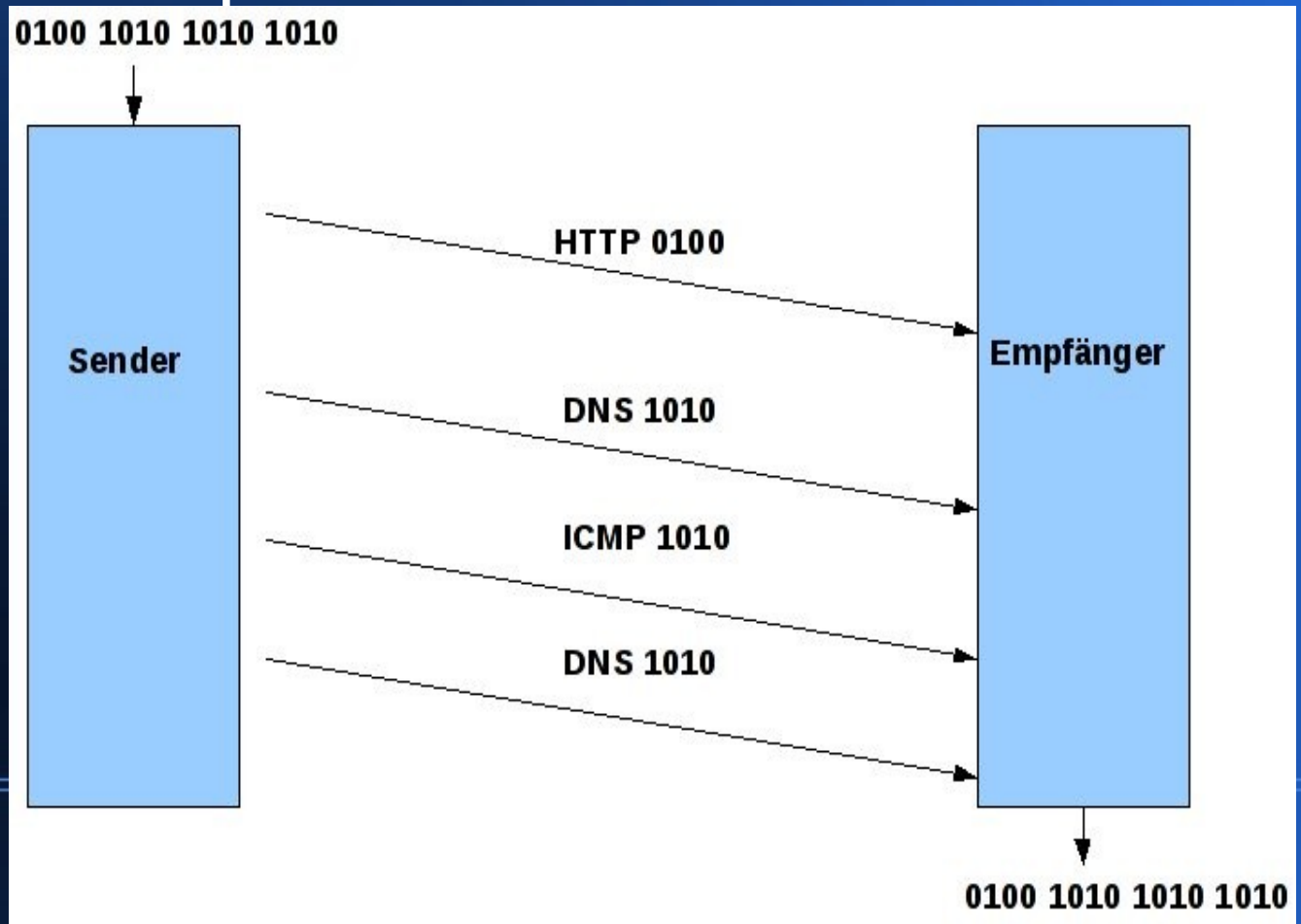
- Network Pump vs. Timing Channel
- Anomaly Detection, Stream Profiling
- Active Wardens (modifizieren verdächtigen Traffic), etwa Normalisierung von Flags
  - Bspw. OpenBSD pf scrub (minimal!)
    - IP ID ersetzen
- → eingeschränkte Nutzungsmöglichkeiten!

Teil II

Protocol Hopping Covert Channels

Protocol Hopping Covert Channel

- PHCC = CC mit Wechselfähigkeit des Netzwerkprotokolls



PHCC (2)

- Ziel: Detektion und Aufdeckung übertragenen Inhalts erschweren
- Erste Implementierung
  - LOKI2
    - Phrack Mag. Vol. 7/51, 1997
    - Autor: ”daemon9”

LOKI2

- Tunneling via UDP und ICMP
- Manueller Protokollwechsel via '/swapt'
  - *”Swapping protocols is broken in everything but Linux. (…) This is why this feature is 'beta'.”* (LOKI2-Artikel im Phrack Mag.)

LOKI2 (2)

swendzel@steffenmobile: ~

```
#define SWAP_T      "/swapt"        /* Swap protocols  */
...
    if (signal(SIGUSR1, swap_t) == SIG_ERR)
        err_exit(1, 1, verbose, L_MSG_SIGUSR1);
...

void d_parse(u_char *buf, pid_t pid, int ripsock) {
    ....
    if (!strncmp(buf, SWAP_T, sizeof(SWAP_T) - 1))
    {
        if (kill(getppid(), SIGUSR1))
            err_exit(1, 1, verbose,
                "[fatal] could not signal parent");
        clean_exit(0);
    }
...
void swap_t(int signo) {
    ...
    close(tsock);

    prot = (prot == IPPROTO_UDP) ? IPPROTO_ICMP : IPPROTO_UDP;
    if ((tsock = socket(AF_INET, SOCK_RAW, prot)) < 0)
        err_exit(1, 1, verbose, L_MSG_SOCKET);
    pprot = getprotobynumber(prot);
    sprintf(buf, "lokid: transport protocol changed to %s\n",
        pprot -> p_name);
    fprintf(stderr, "\n%s", buf);
```

24,1 Anfang

phcct

- = ”protocol hopping covert channel tool” ('07)
- Protokollwechsel jetzt
  - transparent
  - randomisiert
- Zusätzl. Ziel: forensische Analyse erschweren

phcct (2)

- Pakete mit IDs zur späteren Re-Sortierung auf Empfangsseite
  - Features wie Reliability sind für CC nicht neu
    - Implementierung etwa in 'icmptunnel'
- Detektion durch ID möglich, wenn Wert bspw. an unüblicher Stelle inkrementiert wird.
  - Etwa: Mit der Zeit anwachsende IPv4 TTL eingehender Pakete von Host X

Anpassungsfähige CC

- Zwei-Phasen-Kommunikationsprotokoll durch [YADALI08] eingeführt.
  - Network Environment Learning Phase
    - Protokoll-Schnittmengen bilden
    - Protokolle, die nicht geroutet werden oder geblockt werden: abziehen
  - Communication Phase
    - Kommunikation wird abgewickelt
    - Parallel läuft NEL-Phase

- [YADALI08] F. Yarochkin, S.-Y. Dai, C.-H. Lin, Y. Huang, S.-Y. Kuo: Towards Adaptive Covert Communication System, Dep. of Electrical Engineering, National Taiwan University, 2008

Teil 3

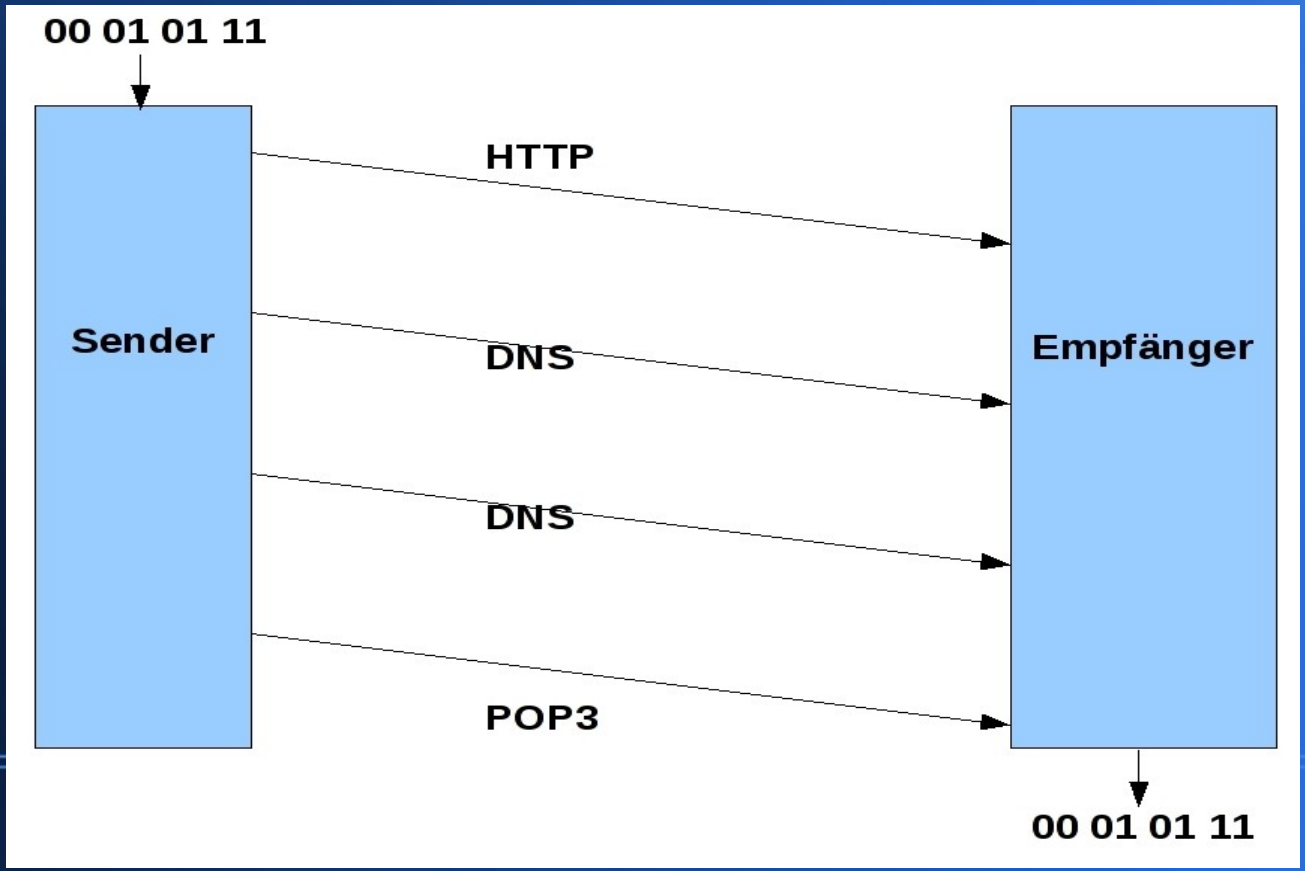
Protocol Channels

Protocol Channels (PC)

- Grob:
  - Informationsübertragung ausschließlich durch Wechsel des Kommunikationsprotokolls
- Genau:
  - *”Ein Protocol Channel ist ein Storage Channel, bei dem Daten **ausschließlich** durch die Information des verwendeten Protokolls gesendeter Pakete übertragen werden **(1)**. Ein Protocol Channel enthält **keine** statischen Identifikationsmerkmale **(2)**. Bei den verwendeten Protokollen muss es sich um für das jeweilige Netzwerk typische, d.h. unauffällige, Protokolle handeln **(3)**.” (Vgl. DA)*

Protocol Channels (2)

- Eindeutige Zuordnung von Bits zu Protokoll
- Etwa HTTP → '00', DNS → '01', ICMP → '10', POP3 → '11'



Protocol Channels (3)

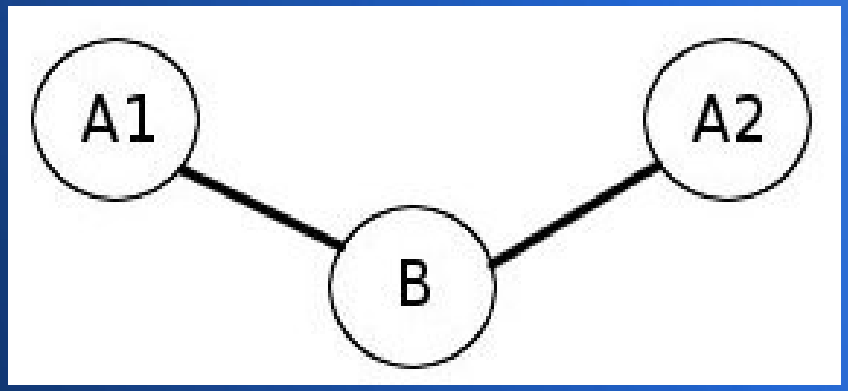
- Kaum detektierbar, da ”normale” Pakete (ohne etwa manipulierte Cookie-Werte) verschickt werden (etwa typischer DNS-Request).
- Letztlich nur Auswertung der Protokoll-IDs
  - ”Ether-Type” bei Ethernet Frame
  - ”Protocol” bei PPP und Ipv4
  - ”Next Header” bei Ipv6
  - Src/Dst Port bei UDP/TCP
  - ...

Protocol Channels (4)

- Verschiedene Layer nutzbar
  - Auch gleichzeitig (d.h. für *einen* Protocol Ch.)
- ”Indirekter Empfang” der Information
  - Bspw. SOCK_STREAM in Application-Layer, aber eigentliche Information in TCP Ziel-Port.
- Proof of Concept Implementierung: ”pct”
- Doch ...

Protocol Channels (5)

- Kein Platz für Korrekturinformationen
- Keine Bestätigungspakete sinnvoll möglich
  - Two Army Problem



- Störung durch eigentlichen Traffic → De-Sync!
- Fehlererkennung schwierig
  - PoC ”pct” nutzt Parity Bit

Bildquelle: Wikipedia: http://de.wikipedia.org/wiki/Datei:2_armeen_problem.jpg

Protocol Channels (6)

- Weniger schwierig: Fragmentierung
- X ”normale” Echo-Requests fallen irgendwann auch auf
  - ”pct” nutzt minimalisiertes 5-Bit-ASCII um Trafficaufkommen zu reduzieren
- Letztlich sind alle Covert Channel-Tools lokal detektierbar; sei es über Listen-Sockets oder andere Veränderungen im User- und Kernelspace.

Weitere Informationen

- Meine Diplomarbeit, andere Veröffentlichungen zum Thema und die Proof of Concept Codes gibt es auf wendzel.de.
- Fragen, Anregungen, konstruktive Kritik:
  - steffenwendzel (at) gmx (dot) net