Malware Trends

Christian J. Dietrich
dietrich [at] internet-sicherheit . de

Institut für Internet-Sicherheit
https://www.internet-sicherheit.de
FH Gelsenkirchen

Agenda

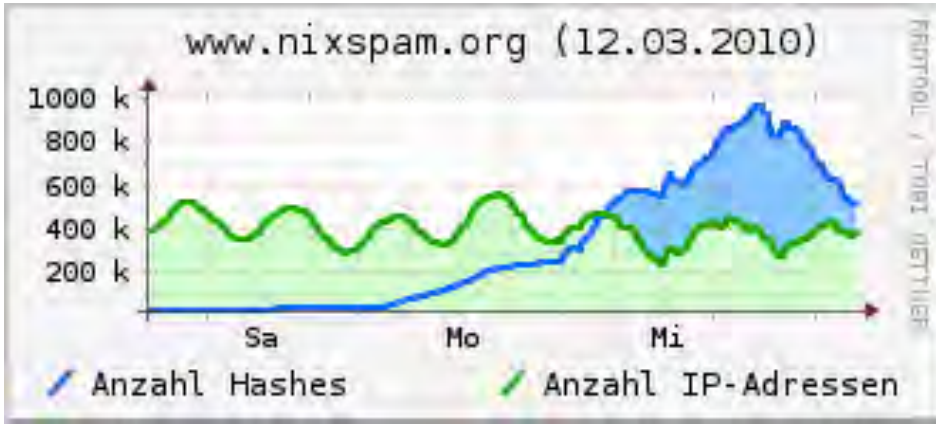
- Einleitung
- Aktuelle Malware-Entwicklung
- Botnetz-Entwicklung
- Ausblick & Herausforderungen
- Fazit

Agenda

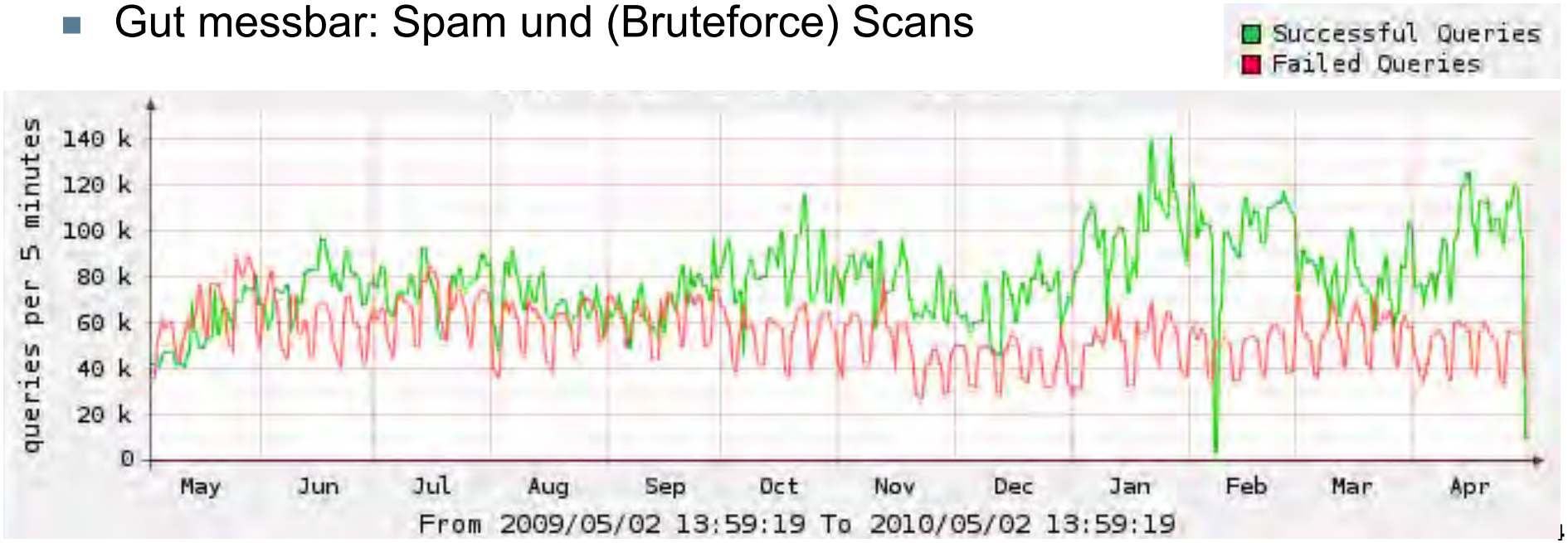
- **Einleitung**
- Aktuelle Malware-Entwicklung
- Botnetz-Entwicklung
- Ausblick & Herausforderungen
- Fazit

Einleitung

- Zunehmendes Malware-Problem
  - 35.000+ „unique“ Samples pro Tag
  - Binary packing
- Effekte nehmen zumindest nicht ab
  - Gut messbar: Spam und (Bruteforce) Scans



Quelle: hostblogger.de



Agenda

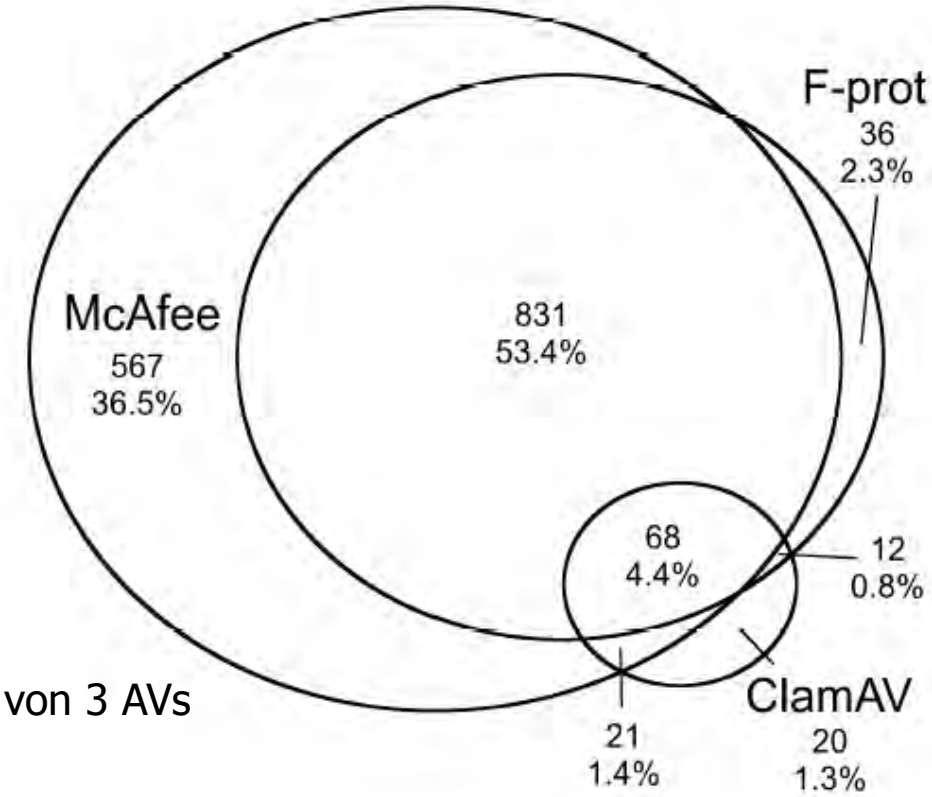
- Einleitung
- **Aktuelle Malware-Entwicklung**
- Botnetz-Entwicklung
- Ausblick & Herausforderungen
- Fazit

Aktuelle Entwicklungen

- Toolkits (Zeus, SpyEye)
- Dezentralisierung von C&C
  - Peer2Peer
  - FastFlux
- Verschlüsselung von C&C
  - Z.B. Virut
- Nomenklatur / Malware-Familien

Quelle: http://www.symantec.com/connect/de/blogs/spyeye-bot-versus-zeus-bot

Malware Label Inconsistency



SDBot family Labels von 3 AVs

Quelle: M. Bailey, Automated Classification and Analysis of Internet Malware

Current Malware Behavior

Observed Behavior	Percentage of Samples	Percentage of Clusters
Installation of a Windows kernel driver:	3.34%	4.24%
Installation of a Windows service:	12.12%	7.96%
Modifying the hosts file:	1.97%	2.47%
Creating a file:	70.78%	69.90%
Deleting a file:	42.57%	43.43%
Modifying a file:	79.87%	75.62%
Installation of an IE BHO:	1.72%	1.75%
Installation of an IE Toolbar:	0.07%	0.18%
Display a GUI window:	33.26%	42.54%
Network Traffic:	55.18%	45.12%
Writing to stderr:	0.78%	0.37%
Writing to stdout:	1.09%	1.04%
Modifying a registry value:	74.59%	69.92%
Creating a registry key:	64.71%	52.25%
Creating a process:	52.19%	50.64%

Quelle: U. Bayer, A View on Current Malware Behaviors, LEET'09

Agenda

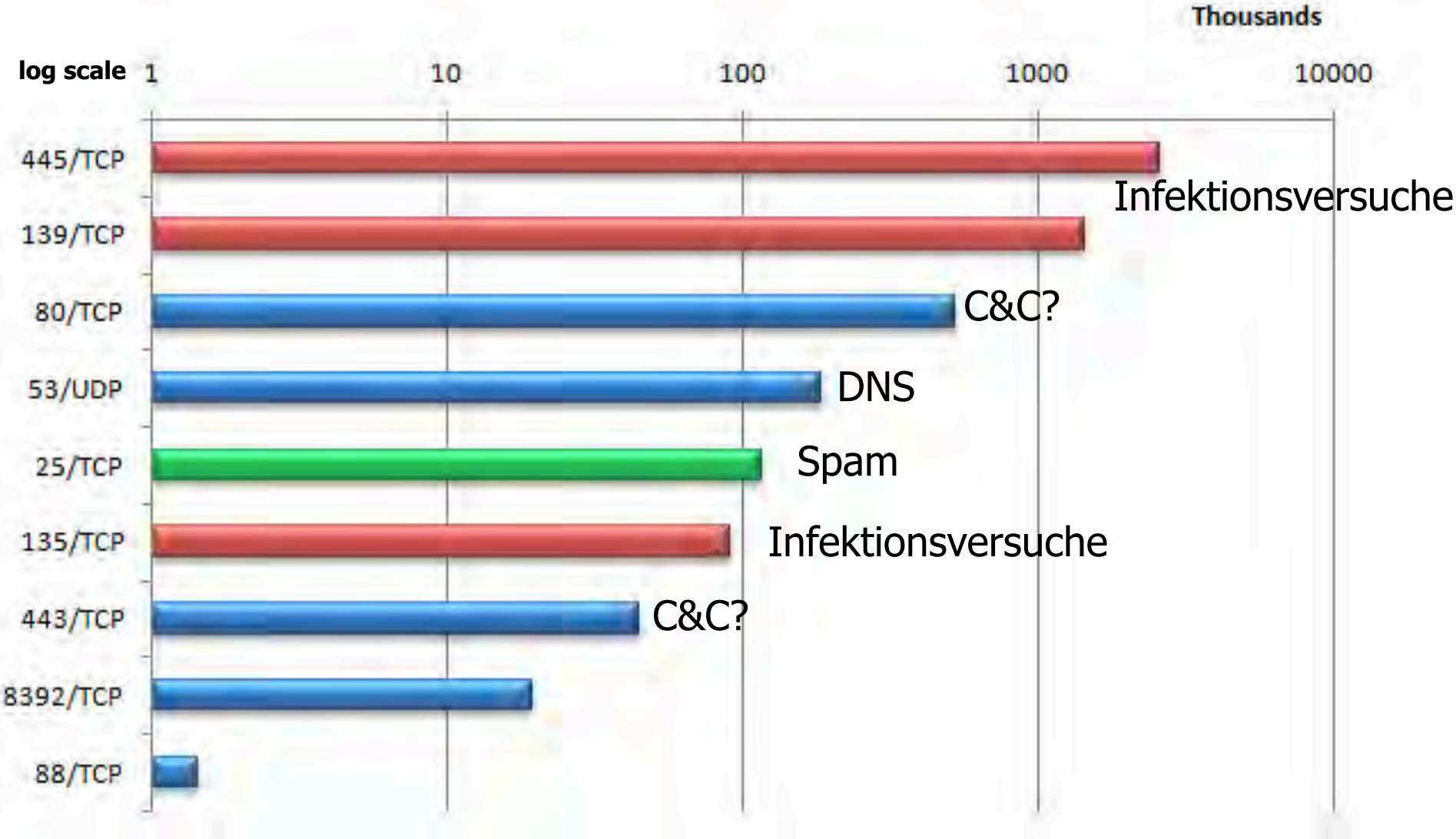
- Einleitung
- Aktuelle Malware-Entwicklung
- **Botnetz-Entwicklung**
- Ausblick & Herausforderungen
- Fazit

Botnetz-Evolution

Evolutionsstufe	Erkennungsmerkmal
IRC-Bots	dport == 6667
IRC auf einem anderen Port	Packet Inspection
HTTP statt IRC	Signatur

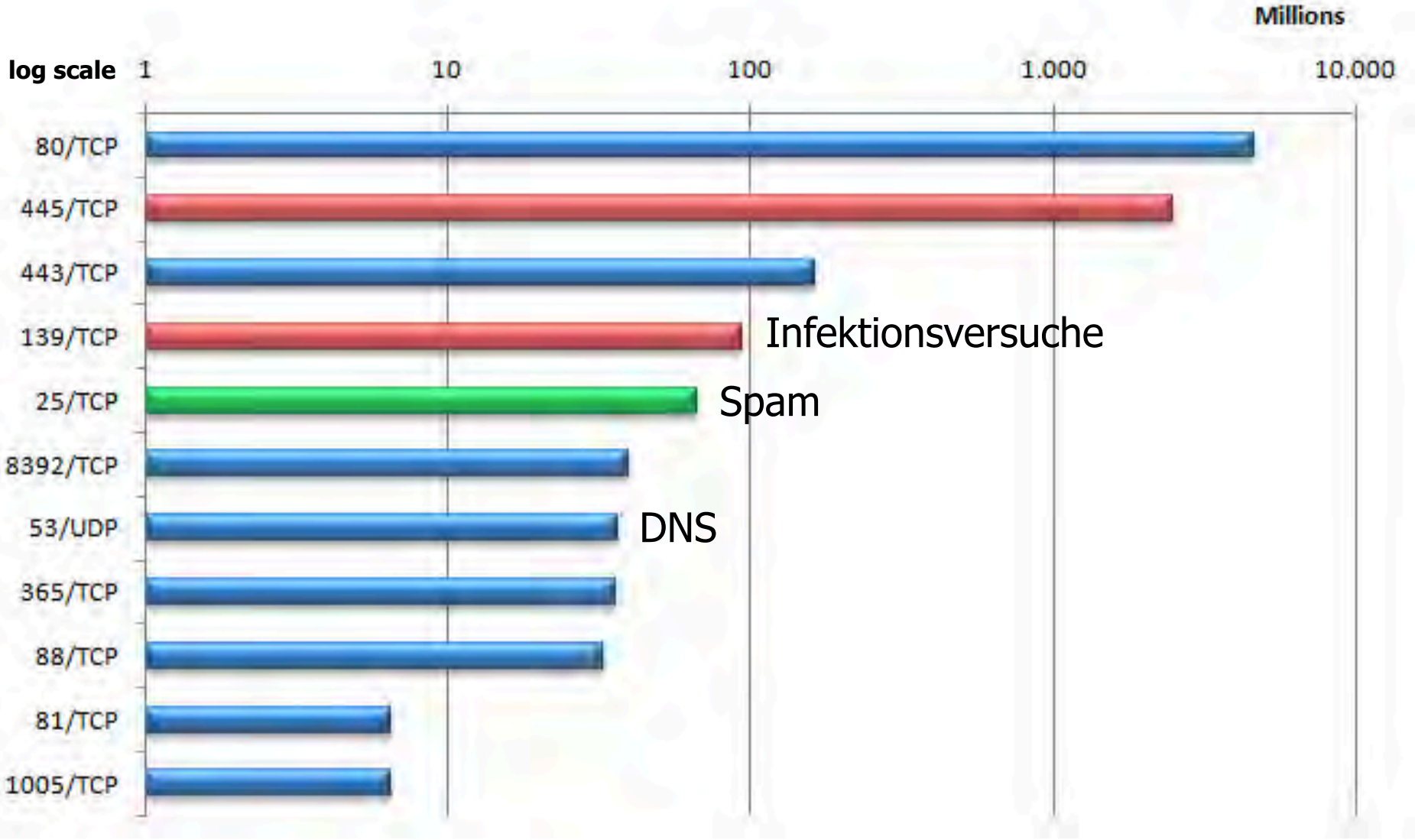
Current Malware Network Behavior

- Verteilung von Destination Ports in reinem Botverkehr (by flows)



Current Malware Network Behavior

- Verteilung von Destination Ports in reinem Botverkehr (by volume)



Protokollverteilung von Botverkehr

dport	#flows	%TLS	%SMTP	%DNS	%HTTP	%Flash	%IRC
135	5823152	0	0	0	0	0	0
445	1039812	0	0	0	0	0	0
53	985804	0	0	100	0	0	0
139	631819	0	0	0	0	0	0
80	257944	0	0	0	95,49	0,48	2,72
6667	156070	0	0	0	0	0	92,76
8392	38459	0	0	0	0	0	0
137	35794	0	0	0	0	0	0
443	35334	12,96	0	0	0	0	0
8800	13604	0	0	0	0	0	0
8069	11888	0	0	0	0	0	100
6668	9610	0	0	0	0	0	100
1867	2310	0	0	0	0	0	97,32
88	1826	0	0	0	100	0	0
5000	1652	0	0	0	0	0	0
5190	1612	0	0	0	0	0	95,91
6271	1344	0	0	0	100	0	0
7000	948	0	0	0	0	0	100
4444	942	0	0	0	0	0	0
1847	823	0	0	0	100	0	0
1239	803	0	0	0	0	0	0
6327	789	0	0	0	99,87	0	0
3009	789	0	0	0	100	0	0
6556	783	0	0	0	0	0	100
65520	763	0	0	0	0	0	73,79
7478	718	0	0	0	100	0	0

- Protokolle auf Non-Std-Ports
- insb. HTTP und IRC
- 443/TCP nur zu **13%** gültiges TLS/SSL
  - Typischerweise „benutzerspezifische“ Protokolle

Botnetz-Evolution

Evolutionsstufe	Erkennungsmerkmal
IRC-Bots	dport == 6667
IRC auf einem anderen Port	Packet Inspection
HTTP statt IRC	Signatur
Obfuskieren/Verschlüsselung	Entropie, zentraler C&C-Server
Replikation/Dezentralisierung (Fastflux/P2P)	Netzwerkverhalten/Anomalien
Netzwerkverhalten randomisieren	Schadfunktion (Spam, DDoS, Honeypots)
Stealth botnet (Ghostnet)	

Signaturbasierte Botnetzerkennung

- Charakteristische Protokollteile ermitteln und als Signaturmerkmal heranziehen

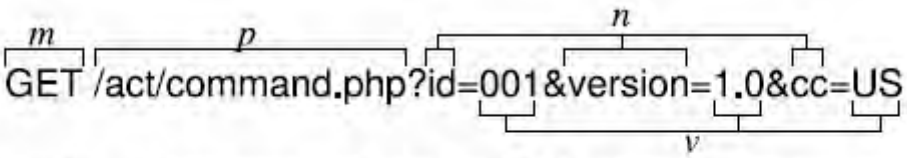


Figure 2: Structure of an HTTP request used in fine-grained clustering. *m*=Method; *p*=Page; *n*=Parameter Names; *v*=Parameter Values.

Quelle: Perdisci, Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces, 2010

➔ Durch **Verschlüsselung** oder Verschleierung auszuhebeln

➔ z.B. Storm, Virut, Waledac, Conficker (static XOR key)

Verschlüsselter C&C

- Schlüssel muss den Kommunikationspartnern bekannt sein
  - Bot (z.B. statisch im Binary)
  - C&C-Server
- Krypto-Verfahren
  - Gleicher Plaintext sollte unterschiedlichen Ciphertext erzeugen
  - Einfache XOR-Verfahren garantieren dies nicht
- Direkt beim ersten Kontakt mit dem C&C-Server nutzen (kein Key Agreement)

Verschlüsselter C&C: Virut

- Ca. 3 Jahre im Umlauf
- IRC als C&C-Protokoll
- Stromverschlüsselung mit **zufälligem** 4 Byte Session Key
  - Verschlüsselung wird bereits beim ersten Kontakt zum C&C-Server genutzt
  - XOR, rotiert Session Key alle 4 Byte, multipliziere den Session Key mit 13

➔ **Woher kennt der Server den Key zum Entschlüsseln?**

Verschlüsselter C&C: Virut

- Ca. 3 Jahre im Umlauf
- IRC als C&C-Protokoll
- Stromverschlüsselung mit **zufälligem** 4 Byte Session Key
  - Verschlüsselung wird bereits beim ersten Kontakt zum C&C-Server genutzt
  - XOR, rotiert Session Key alle 4 Byte, multipliziere den Session Key mit 13

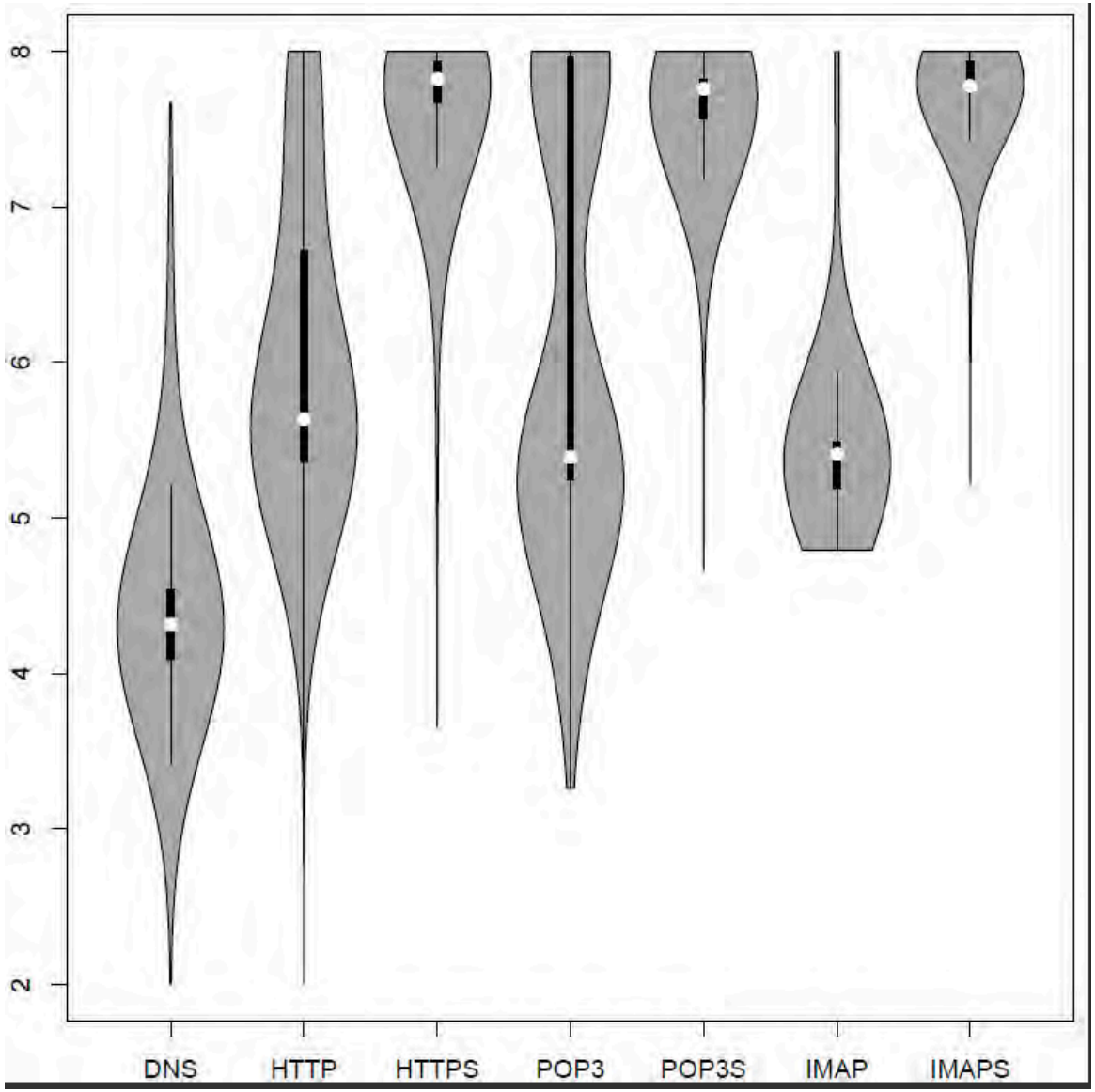
➔ **Woher kennt der Server den Key zum Entschlüsseln?**

- Möglicherweise:
  Known plaintext-attack auf das eigene Protokoll

Quelle: http://www.secureworks.com/research/threats/virut-encryption-analysis/

Entropie-basierte Erkennung von verschlüsseltem C&C

- Protokolle zeigen eine typische Verteilung der Entropie
- Verschlüsselte Daten haben eine hohe Entropie



Welche Bots nutzen verschlüsselten C&C?

botnet family	*# samples*	*# enc.*	*% enc.*
VIRUT	204	192	94.12
RBOT	101	93	92.08
ZHELATIN	41	34	82.93
SALITY	85	62	72.94
SPYGAMES	139	101	72.66
ROTATOR	300	168	56.00
CASINO	140	11	7.86
POISON	26	1	3.85
VIKING_DLL	158	2	1.27

Flow-basierte Botnetzerkennung

- Keine klassischen Signaturen
- Verhaltensanalyse auf Netzwerk-Ebene
  - Ist der Netzwerkverkehr verschlüsselt?
  - Welches Payload-Protokoll wird (vermutlich) verwendet?
  - Kennwerte
    - Flows pro Zeiteinheit
    - Pakete pro Flow
    - Bytes pro Paket
    - Bytes pro Zeiteinheit

➔ **resistent gegen verschlüsseltes C&C**

Forschung Botnetz-Erkennung

- Ziel: Erkennung von modernen Botnetzen
  - Einsatz von Verschlüsselung
  - Dezentralisierung
  - Replikation
- Vorteile der Verhaltensanalyse auf Netzwerk-Ebene
  - Datenschutzfreundlich
    - Kein Speichern von Payload oder IP-Adressen
  - Protokollunabhängig
  - Unabhängig vom Auftreten der Schadfunktion
- 2 Hochschulen: Prof. Thorsten Holz (HGI, RUB), if(is) FH GE
- **Wir suchen insbesondere ISPs als Projektpartner!**

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

Christian J. Dietrich
dietrich [at] internet-sicherheit . de

Institut für Internet-Sicherheit
https://www.internet-sicherheit.de
FH Gelsenkirchen