# Honeypots, Darknets und Datenanalyse

Michael Gröning

# Gliederung

1 Einleitung

2 Honeypots und Darknets
- Honeypots
- Darknets
- Synthese aus Honeypot und Darknet

3 Datenanalyse
- Verhaltensbasierte Analyse
- Entropie-Analyse
- Weitere mögliche Analysetechniken

# Gliederung

1 Einleitung

2 Honeypots und Darknets
- Honeypots
- Darknets
- Synthese aus Honeypot und Darknet

3 Datenanalyse
- Verhaltensbasierte Analyse
- Entropie-Analyse
- Weitere mögliche Analysetechniken

# Gliederung

1 Einleitung

2 Honeypots und Darknets
- Honeypots
- Darknets
- Synthese aus Honeypot und Darknet

3 Datenanalyse
- Verhaltensbasierte Analyse
- Entropie-Analyse
- Weitere mögliche Analysetechniken

# Einleitung

## Kurze Vorstellung

- Studium der Technischen Informatik(B.Sc.)
- Arbeit beim DFN-CERT seit 2006
- Incident Response Team - Aufgabe: Schwachstellenanalyse und Bewertung
- Bachelorarbeit zum Thema Malware-Detection mit Honeypots

# Einleitung

## Inhalte dieses Vortrages

- kurze Einführung in Aktuelle Techniken der IT-Sicherheitsforschung
- Im letzten Teil des Vortrages: Kurze Demo von verschiedenen Analysemethoden

# Einleitung

Bitte zögern Sie nicht mit Fragen!

# Einführung

## Was ist ein Honeypot?

# Einführung

## Was ist ein Honeypot?

Definition aus der Securityfocus-Mailingliste[1]:

> *A honeypot is an information system resource whose value lies in unauthorized or illicit use of the resource.*

Frei übersetzt:

> *Ein Honeypot ist eine Informationstechnische Ressource, deren Wert in ihrem Missbrauch besteht.*

[1] www.securityfocus.com/archive/119

# Einführung

## Was ist ein Honeypot?

Honeypots sind dazu da, dass sie angegriffen werden.

- Ermöglichen die Analyse unbekannter Angriffe
- Ermöglichen die Überwachung eines Angreifers und Verhaltensanalyse
- Dienen dem Erkenntnisgewinn über neue Angriffstechniken

# High-Interaction vs. Low-Interaction

## verschiedene Typen von Honeypots

### High-Interaction

- Bilden komplettes System nach, inklusive Betriebssystem und anderer erforderlicher Dienste
- Bieten die Möglichkeit auch bisher unbekannte Arten von Exploits zu detektieren.

### Low-Interaction

- LI-Honeypots simulieren meist nur einen angreifbaren Service (HTTP,SMB,RPC...)
- Sie ermöglichen Analysen über die Häufigkeit von Angriffen

# Beispiel: Argos

## High-Interaction Honeypot

- Basiert auf QEMU Virtualisierungsumgebung und unterstützt alle x86 Betriebssysteme
- Erkennt Bufferoverflows an Veränderungen in der simulierten CPU

# Beispiel: Honeytrap

## Low-Interaction Honeypot


- ab 2005 von Tillmann Werner am BSI[2] entwickelt
- mehrere neue Ansätze für Low Interaction Honeypots
- Betrieb als reiner Serverdienst im Usermodus

[2] Bundesamt für Sicherheit in der Informationstechnik

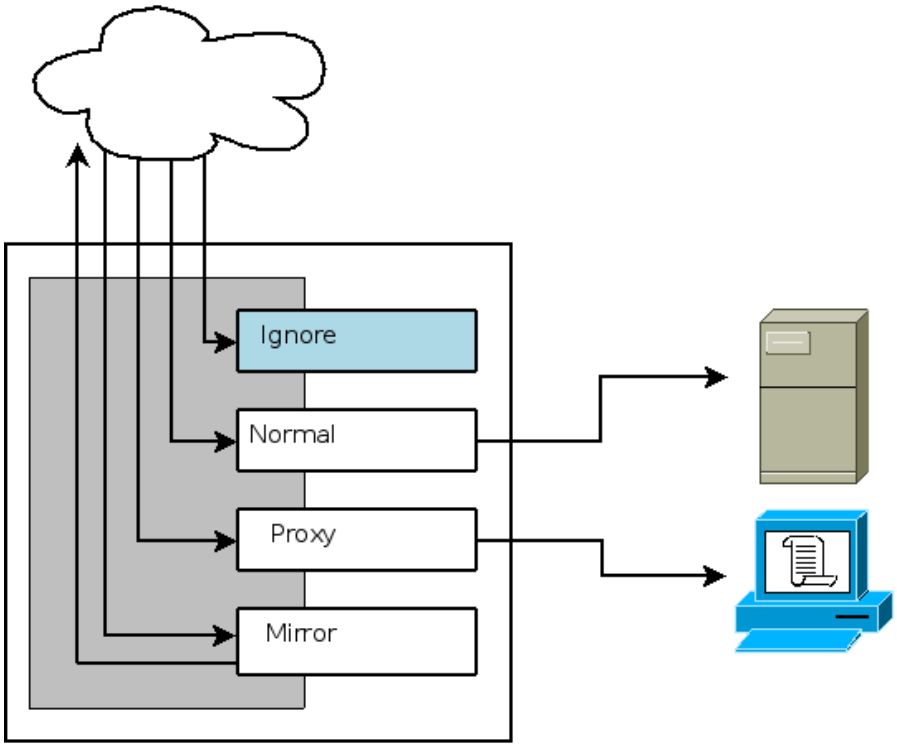
# Beispiel: Honeytrap

## Unterschiede zu bestehenden LI-Honeypots

- Verrschiedene Betriebsmodi, die auch kombiniert werden können
- Honeytrap enthält keine Schwachstellenmodule.
- Schickt generische Antworten auf alle Requests.

# Beispiel: Honeytrap

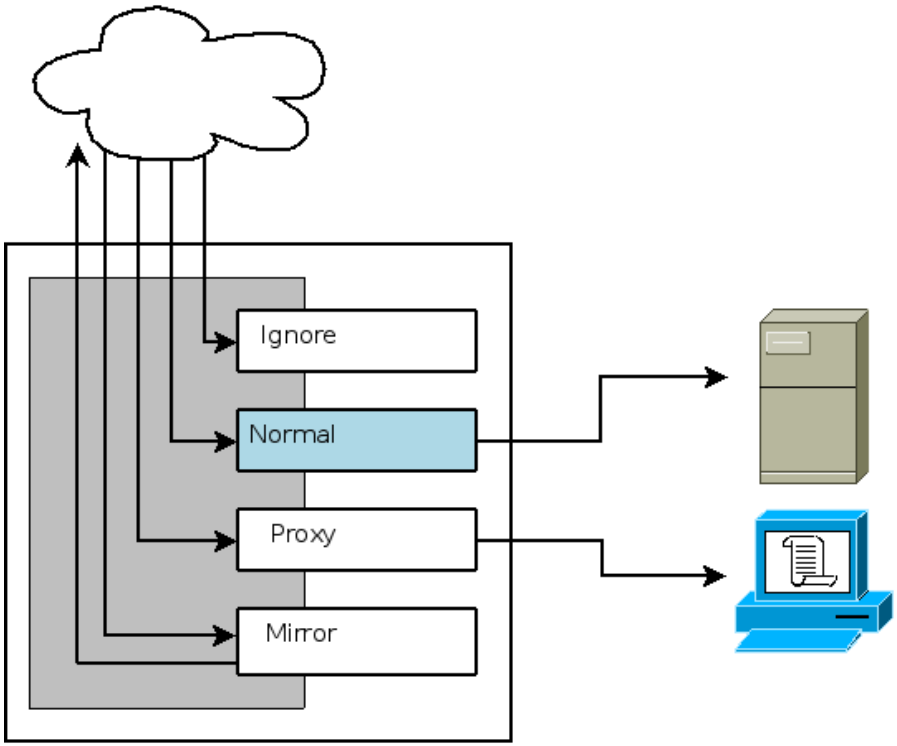
## Betriebsmodus Ignore




Ankommende Requests werden durch Honeytrap ignoriert und an das lokale Betriebssystem weitergeleitet.

# Beispiel: Honeytrap

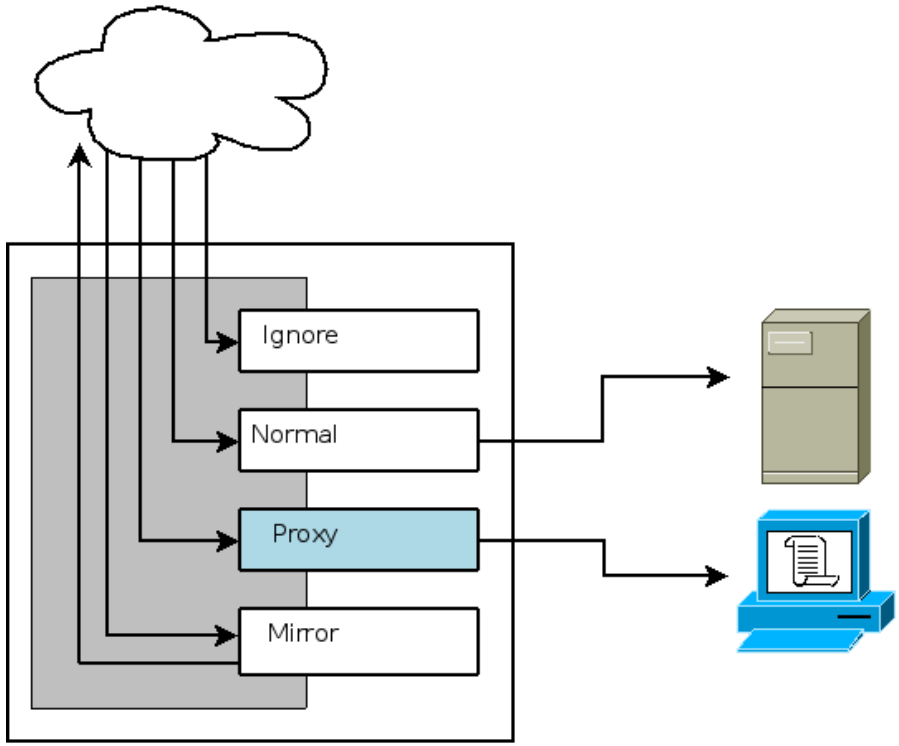
## Betriebsmodus Normal




Ankommende Requests werden rudimentär beantwortet, die Daten werden gespeichert.

# Beispiel: Honeytrap

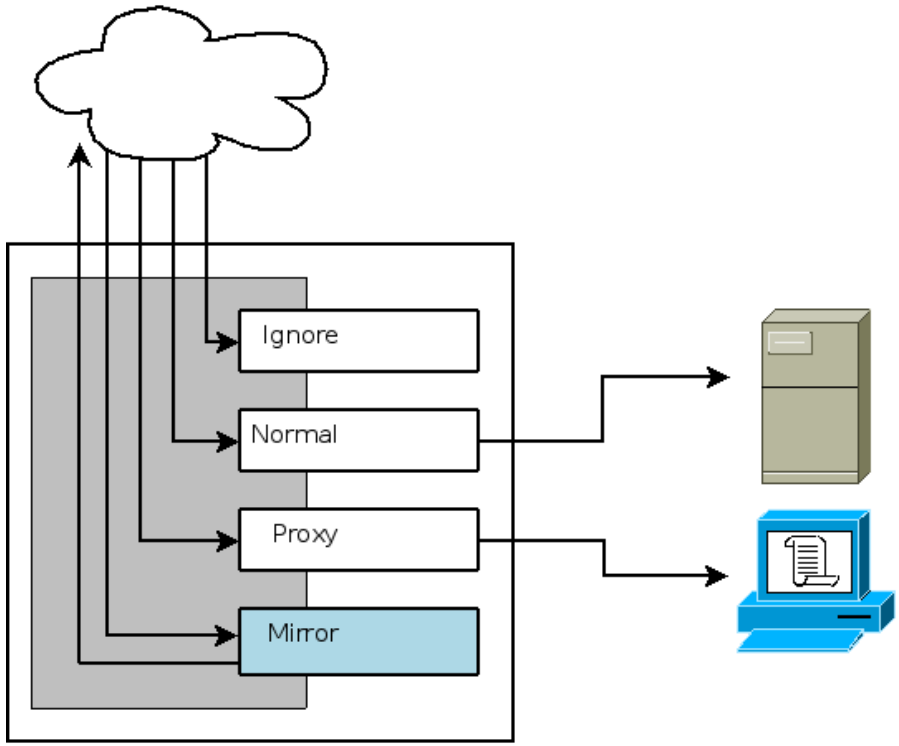
## Betriebsmodus Proxy




Honeytrap leitet Requests an einen Anderen Rechner weiter.
(z.B. ein HI-Honeypot)

# Beispiel: Honeytrap

## Betriebsmodus Mirror




Ankommende Requests werden an den vermeintlichen Angreifer zurückgespiegelt.

# Darknets

Was ist ein Darknet?

# Darknets

## Einführung

Darknets sind Netzbereiche in denen keinerlei reguläre Dienste angeboten werden, d.h. leere IP-Addressbereiche mit denen keine Rechner verbunden sind

# Darknets

## Beispiel


Quelle: http://xkcd.com/195/ ©by Randall Munroe

# Darknets

## Wozu sind Darknets zu gebrauchen?

- Keine angebotenen Dienste
- kein regulärer Netzwerkverkehr
- Das Hintergrundrauschen tritt viel deutlicher zu Tage als in anderen Netzbereichen.

# Darknets

Hintergrundrauschen?

# Darknets

## Was ist Hintergrundrauschen?

Im Internet gibt es immer einen gewissen Anteil an Hintergrundrauschen.

- Fehlkonfigurierte Rechner,Router usw.
- Traffic an abgeschaltete Systeme

aber auch:

- Botangriffe!
- Peer2Peer-Verkehr!

Dahinter sind wir her!

# Caida Network Telescope

## Beispiel

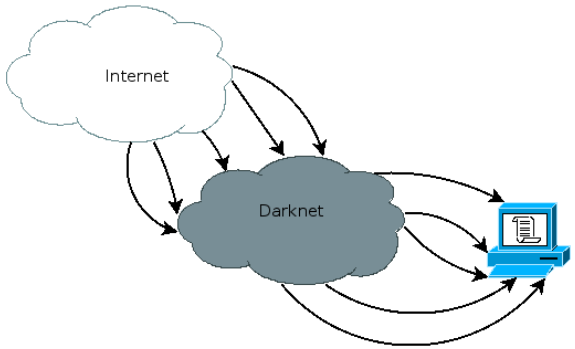

- 16,7 Millionen IP-Adressen
- empfängt etwa 0,4 Prozent aller Botnetzangriffe
- durch die Größe bedingte, extrem schnelle Reaktionszeit

http://www.caida.org/research/security/telescope/

# Wie können diese Ansätze kombiniert werden?

# Honeytrap als Meta-Honeypot im Darknet

## Synthese aus Honeypot und Darknet




Idee: Man bindet eine große Anzahl von IP-Adressen aus dem Darknet an einen einzigen Low-Interaction Honeypot.

- Theoretisch schnelles Registrieren neuer Angriffe möglich
- Sehr viele Angriffe in kurzer Zeit
- Theoretisch hohe Skalierbarkeit (>1000 IP-Adressen)

Mehr dazu in 6 Monaten ;-)

# Teil 2: Datenanalyse

## Übersicht

- Verhaltensbasierte Analyse - LibEmu
- Entropie-Analyse
- Weitere mögliche Analysetechniken

# Einführung

Low Interaction Honeypots bieten keine Möglichkeit das Verhalten von Malware an einem voll funktionsfähigen Rechnersystem zu analysieren.
Dies erfordert die Entwicklung spezieller Analysetechniken

# Verhaltensbasierte Analyse

## Verhaltensbasierte Analyse

# Verhaltensbasierte Analyse -libEmu

## Einführung

- Entwicklung ab 2005 von Paul Baecher und Markus Kötter
- Neuer Ansatz: Nachbau einer CPU und Ergänzung um Elemente einer simulierten Windows-API
- existiert als C-Bibliothek und kann deshalb in neue Projekte eingebunden werden. (z.B. in Honeytrap)

http://libemu.carnivore.it

# Verhaltensbasierte Analyse -libEmu

## libEmu

Kurze Demonstration

# Verhaltensbasierte Analyse -libEmu

## Fazit

- Konzept vereinigt viele der Möglichkeiten eines High-Interaction Honeypots mit der Skalierungsfähigkeit eines Low-Interaction Honeypots
- kann auch unbekannten oder verschlüsselten Shellcode entdecken
- Funktioniert nur mit beschriebenen API-Aufrufen kann aber für viele Weitere APIs bei Bedarf erweitert werden.

# Entropie-Analyse

Entropie-Analyse

# Entropie-Analyse

## Bestimmung des Informationsgehaltes von Honeypotdaten

Frage: welche Eigenschaften besitzt Shellcode?

- Shellcode muss i. Allg. in den Nutzdaten eines Paketes plaziert werden.
- Shellcode besteht meist aus Assemblercode.
- Wird oft von Konstrukten begleitet, welche die Erfolgschancen erhöhen sollen (z.B. NOP-Sleds, keine NULL-Character).

Erwartung: Shellcode hat deutliche strukturelle Unterschiede von der ihn umgebenden Struktur.

# Entropie-Analyse

$$H(x) = -\sum_{i=1}^{n} p(i) \log_2 p(i)$$

Entropie ist definiert als das Maß für den mittleren Informationsgehalt pro Zeichen eines Textes.

# Entropie-Analyse

Was bedeutet das?

# Entropie-Analyse

$$H(x) = -\sum_{i=1}^{n} p(i) \log_2 p(i)$$

Wird ein Bytestring betrachtet, so tritt jedes Zeichen *i* aus dem beschreibenden Zeichenalphabet mit einer bestimmten Wahrscheinlichkeit *p(i)* bezogen auf die Gesamtanzahl der Zeichen auf.
Der Unterschied zwischen der theoretischen maximalen Entropie und der tatsächlichen Entropie bezeichnet man als *Redundanz*.

# Entropie-Analyse

## Beispiele

- Natürlichsprachliche Texte: Bestimmte Zeichen aus dem Alphabet kommen häufiger vor als andere.
- Münzwurf, Würfel: Alle Ergebnisse eines Münzwurfs sollten gleich oft vorkommen. Dies lässt sich anhand des Entropiewertes überprüfen.
- Datenformate: verschiedene Datenformate enthalten unterschiedliche Mengen an Redundanz. Dies ist für bestimmte Datenformate Typisch. (z.B. ASCII vs. PGP)

# Entropie-Analyse

Idee: Beim Vergleich von einfachen Binärdaten mit Shellcode sollten Unterschiede messbar sein.

# Entropie-Analyse

Frage: Reichen diese Unterschiede aus, um Unterscheidungen treffen zu können?

# Entropie-Analyse

## Wie unterscheidet man ungefährliche Daten von Malware-Code?

```
00 30 00 30 00 32 00 20  00 35 00 2e 00 31 00 00  |.0.0.2. .5...1..|
00 00 00 00 00 00 5a ff  53 4d 42 75 00 00 00 00  |......Z.SMBu....|
18 07 c8 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00 ff fe 00 08 30 00 04  ff 00 5a 00 08 00 01 00  |.....0....Z.....|
2f 00 00 5c 00 5c 00 31  00 34 00 31 00 2e 00 39  |/..\.\.1.4.1...9|
00 2e 00 32 00 34 00 30  00 2e 00 32 00 39 00 5c  |...2.4.0...2.9.\|
00 49 00 50 00 43 00 24  00 00 00 3f 3f 3f 3f 3f  |.I.P.C.$...?????|
00 00 00 00 64 ff 53 4d  42 a2 00 00 00 00 18 07  |....d.SMB.......|
```

Ist dieser Binärtext gutartig?

# Entropie-Analyse

## Wie unterscheidet man ungefährliche Daten von Malware-Code?

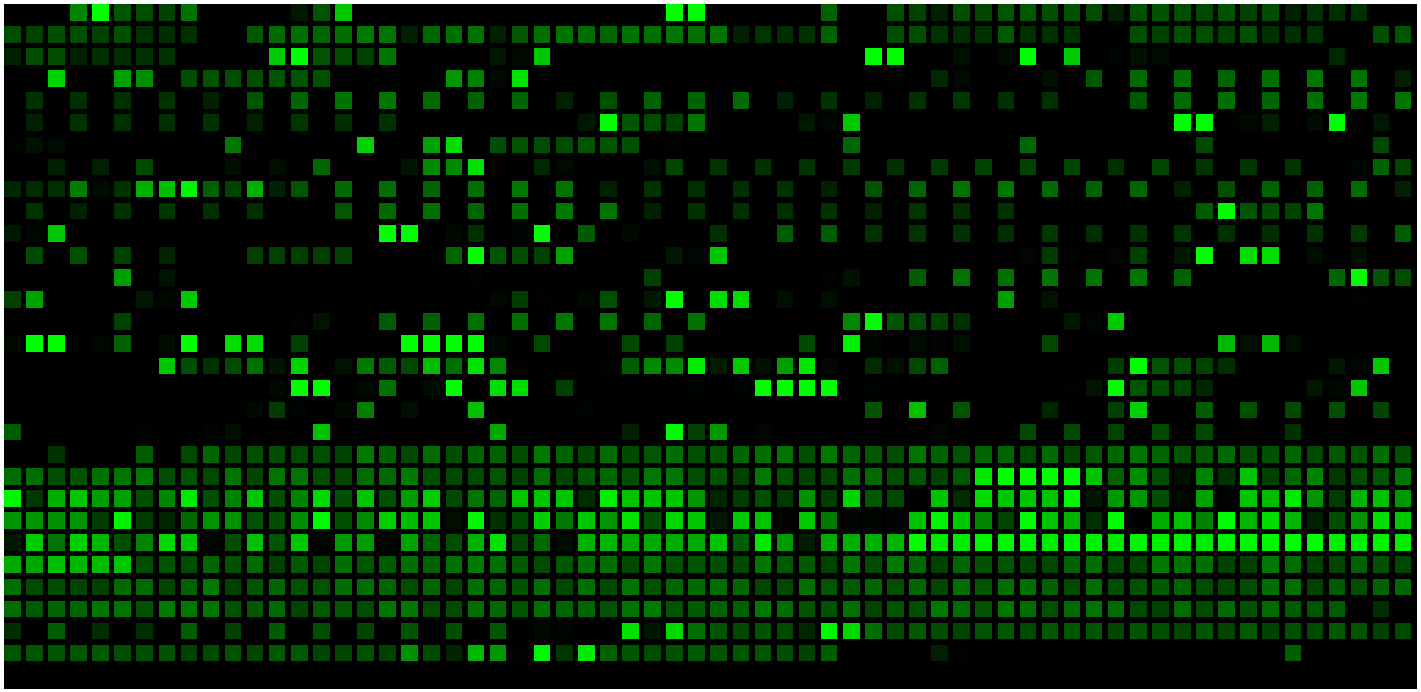
```
51 42 78 52 41 45 6f 69  4f 56 43 57 e8 ff ff ff  |QBxRAEoiOVCW....|
ff c2 5f 8d 4f 10 80 31  c4 41 66 81 39 4d 53 75  |.._.O..1.Af.9MSu|
f5 38 ae c6 9d a0 4f 85  ea 4f 84 c8 4f 84 d8 4f  |.8....O..O..O..O|
c4 4f 9c cc 49 73 65 c4  c4 c4 2c ed c4 c4 c4 94  |.O..Ise...,.....|
26 3c 4f 38 92 3b d3 57  47 02 c3 2c dc c4 c4 c4  |&<O8.;.WG..,....|
f7 16 96 96 4f 08 a2 03  c5 bc ea 95 3b b3 c0 96  |....O.......;...|
96 95 92 96 3b f3 3b 24  69 95 92 51 4f 8f f8 4f  |....;.;$i..QO..O|
88 cf bc c7 0f f7 32 49  d0 77 c7 95 e4 4f d6 c7  |......2I.w...O..|
```

Ist dieser Binärtext bösartig?

Betrachten wir die Entropiewerte

# Entropie-Analyse

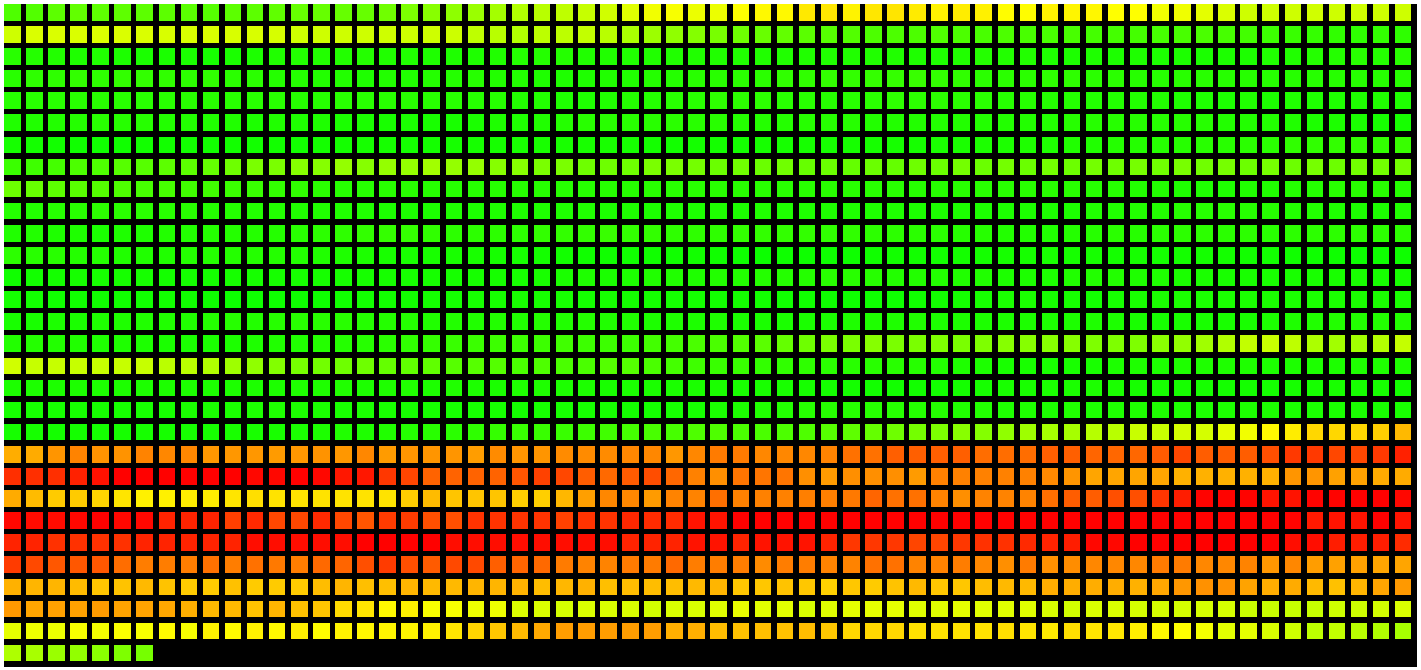
## Beispiel für Shellcode



Darstellung des Bytestreams eines Paketes(0x00 = Schwarz, 0xFF = Hellgrün)

# Entropie-Analyse

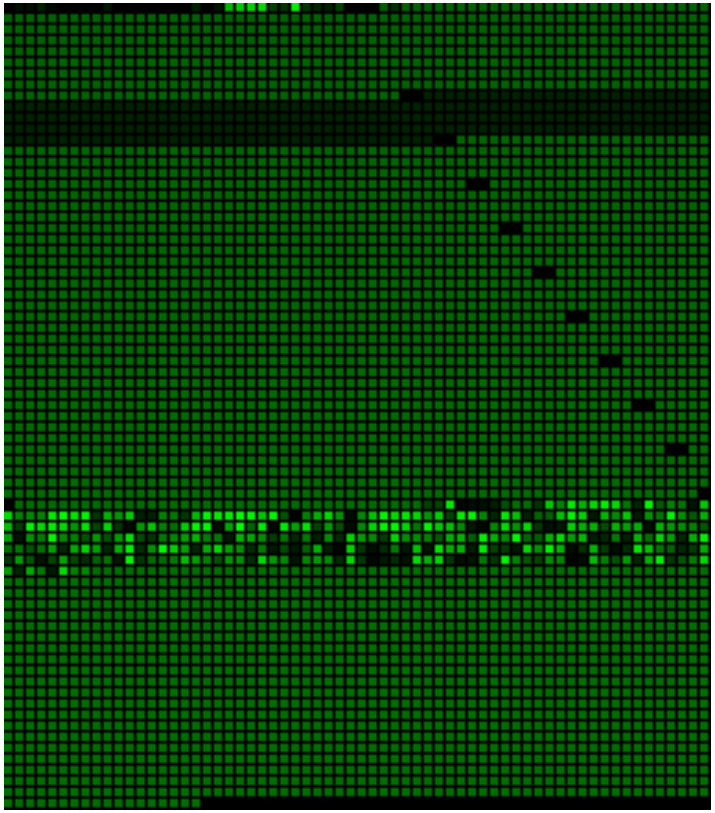
## Beispiel für Shellcode



Darstellung der Entropie mit einem Sliding-Window der Größe 256 Byte

# Entropie-Analyse

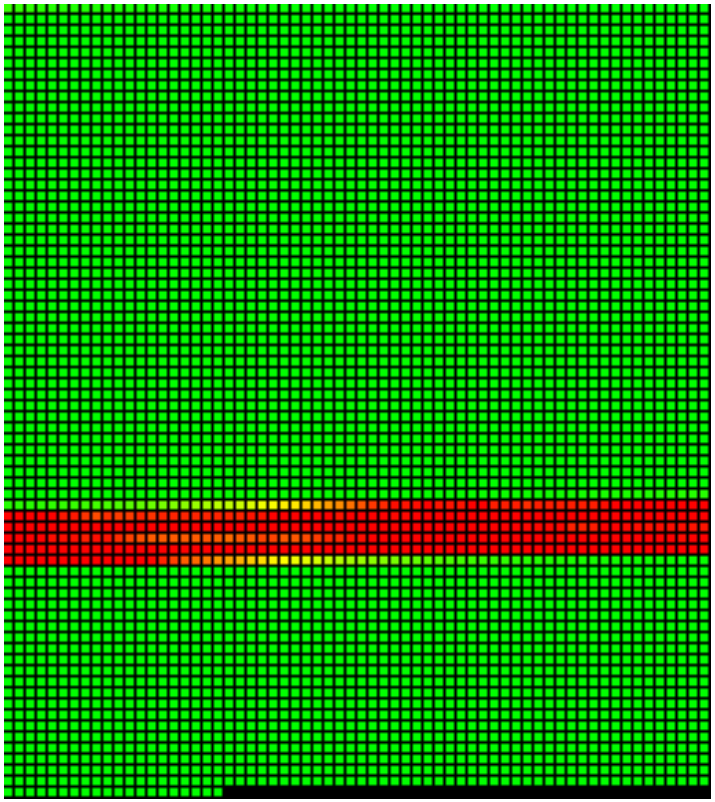
## Beispiel für Shellcode II



Darstellung des Bytestreams eines Paketes(0x00 = Schwarz, 0xFF = Hellgrün)

# Entropie-Analyse

## Beispiel für Shellcode II



Darstellung der Entropie mit einem Sliding-Window der Größe 256 Byte

# Entropie-Analyse

## Fazit

Beim Vergleich zwischen der lokalen Entropie eines Sliding Window mit der durchschnittlichen Entropie des Paketes können signifikante Unterschiede festgestellt werden.

# Entropie-Analyse

## Weitere Analysetechniken

# Scannen nach XOR-Codierten Strings

## Strukturanalyse von Honeypotdaten

Shellcode wird in manchen Exploits mit bestimmten Werten XOR-Verknüpft.

- Unkenntlichmachung von Assembler-Code
- Verschleierung von NULL-Bytes im Code
- Erschwerung von Code-Analyse

# Scannen nach XOR-Codierten Strings

## Strukturanalyse von Honeypotdaten

macht man eine XOR-Verknüpfung der aufeinanderfolgenden Character im String erhält man ein Maß für den die Unterschiede der Character.

```
0x42 XOR 0x43 = 0x01
```

Dabei gehen diese Unterschiede auch nicht verloren wenn die Character mit dem gleichen Byte XOR-Verschlüsselt sind:

```
0x42 XOR 0xC4 = 0x86
0x43 XOR 0xC4 = 0x87
0x86 XOR 0x87 = 0x01
```

# Scannen nach XOR-Codierten Strings

## Strukturanalyse von Honeypotdaten

Idee: Verschiedene Funktionen der Windows-API erwarten Stringketten als Parameter. Vielleicht kann man danach in einem Bytestring suchen.

Beispiel: Übergabe einer URL als ASCII-String

```
'http':
'h'XOR't'=0x1c
't'XOR't'=0x00
't'XOR'p'=0x04
```

```
6e 99 ba 7e 84 b1 1c 00  04 4a 15 00 1e 03 03 1f  |n..~.....J......|
1d 06 1b 1f 04 02 19 1c  03 03 08 03 0d 01 02 18  |................|
```

# Scannen nach XOR-Codierten Strings

## Strukturanalyse von Honeypotdaten

### Kurzer Ausschnitt aus einem Datenpaket:

```
9d 07 a4 66 4e b2 e2 44  68 0c b1 b6 a8 a9 ab aa  |...fN..Dh.......|
c4 5d e7 99 1d ac b0 b0  b4 fe eb eb XX XX XX ea  |.]..............|
XX XX ea XX XX XX ea XX  XX f6 fe XX XX XX XX eb  |................|
a9 a6 be b7 bd c4 4d 53  4d 64 51 69 41 58 46 6e  |......MSMdQiAXFn|
```

### Das ganze nochmal, aber diesmal ohne XOR-Codierung

```
59 c3 60 a2 8a 76 26 80  ac c8 75 72 6c 6d 6f 6e  |Y.`..v&...urlmon|
00 99 23 5d d9 68 74 74  70 3a 2f 2f 31 32 31 2e  |..#].http://XXX.|
XX XX 2e XX XX XX 2e XX  XX XX 3a XX XX XX XX 2f  |XX.XXX.XXX:XXXX/|
6d 62 7a 73 79 00 89 97  89 a0 95 ad 85 9c 82 aa  |XXXXX...........|
```

# Analyse von Metadaten

## Strukturanalyse von Honeypotdaten

Metadaten sind hier bestimmte zusammenhänge wie Portnummern, IP-Addressen, Streamgrößen...
Beispielsweise Benutzung bestimmter bekannter Ports für bestimmte Angriffe oder Datenverbindungen.
z.B. Conficker Peer2Peer-Traffic

# Zusammenfassung

- Honeypots und die Auswertung der gewonnen Daten sind zur Zeit ein sehr aktives Gebiet der Sicherheitsforschung.
- Es werden dauernd neue Angriffe entwickelt. Kreativität ist gefragt!

# Fragen?


KTHXBYE!
KoFoRd