Sondervorlesung „Netz-Sicherheit“: Honeypots - Elektronische Köder im Internet

Tillmann Werner, Edwin Reusch

CERT-Bund, Bundesamt für Sicherheit in der Informationstechnik

Universität Bonn, 7. November 2006

BSI

- Bundesamt für Sicherheit in der Informationstechnik, Bonn
- Drei Fachabteilungen + Verwaltungsabteilung
- Abteilung 1: „Sicherheit in Anwendungen, KRITIS und im Internet“
- Referat 121 - CERT-Bund, „Computer-Notfallteam für Bundesbehörden“

- Aufgaben
  - Veröffentlichen von Advisories zu aktuellen Schwachstellen
  - Monitoring und Bewertung der Bedrohungslage im Internet
  - Incident Handling bei sicherheitskritischen Vorfällen (im Bereich Bund)
  - Anlassbezogene Analyse von Schadprogrammen und Angriffen
  - Point-of-Contact für internationale CERTs

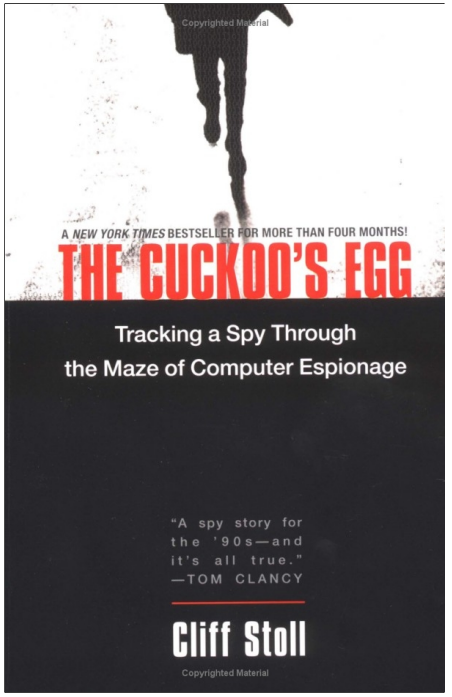
Menu

Apéritif	Clifford Stoll: „The Cuckoo‘s Egg“
Hors d‘oeuvre	Definition und Motivation beim Einsatz von Honeypots
Entrée	Kategorien von Honeypots
Plat Principal	Ausgewählte Tools und Techniken Honeynets der 1., 2. und 3. Generation Honeywall - Data Capture und Data Control
Fromage	Honeypot-Live-Demonstration: honeytrap Angriffsanalyse-Beispiel
Dessert	Spezielle Honeypots Honeypot-Identifikation
Café	Offene Diskussion

The Cuckoo‘s Egg

The Cuckoo‘s Egg, Clifford Stoll, 1990, ISBN 0-7434-1146-3

Stoll setzte in den 80er Jahren am Lawrence Berkeley Laboratory als Systemadministrator als einer der Ersten **Honeypot-Techniken** ein, **um einen Hacker zu überführen**.



Er platzierte **realistische Dateien und Informationen** mit militärischem Hintergrund auf einem System mit **speziellen Überwachungsvorrichtungen**.

Aufgrund der gefälschten Daten konnte der Hacker überführt und außerdem ein ungarischer KGB-Spion identifiziert werden.

Honeypot – Definition

A honeypot is an information system resource whose value lies in unauthorized or illict use of that resource.

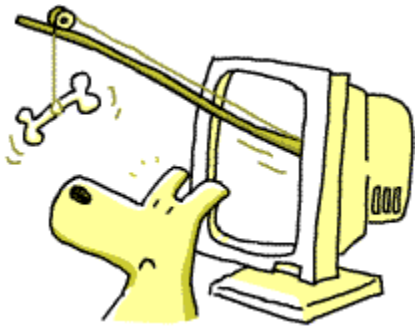
Quelle: „Know Your Enemy“, The Honeynet Project, 2004, Addison-Wesley

A honeypot is a trap set to detect, deflect or in some manner counteract attempts at unauthorized use of information systems.

Quelle: <http://en.wikipedia.org/wiki/Honeypot_(computing)>

Honeypots werden eingesetzt, um von anderen Systemen abzulenken, neue Angriffsmethoden zu identifizieren oder automatisch Gegenmaßnahmen zu ergreifen.

Sie fungieren dabei als Köder und sollen Angreifer mit ihrer Attraktivität anlocken.



Sun Tzu – Über die Kriegskunst

Sun Tzu, 544-496 v.Chr., chinesischer Militär-Stratege.

Er veröffentlichte das Werk „Sun Tsu Ping Fa“
(**Über die Kriegskunst**) mit 13 Kapiteln in Postulaten,
die noch heute viel zitiert werden.

Kapitel 1: Strategische Überlegungen
Kapitel 13: Der Einsatz von Spionen

„All warfare is based on deception.“

Sun Tzu, The Art of War, I - Laying Plans, 18

„Hold out baits to entice the enemy. Feign disorder, and crush him.“

Sun Tzu, The Art of War, I - Laying Plans, 20

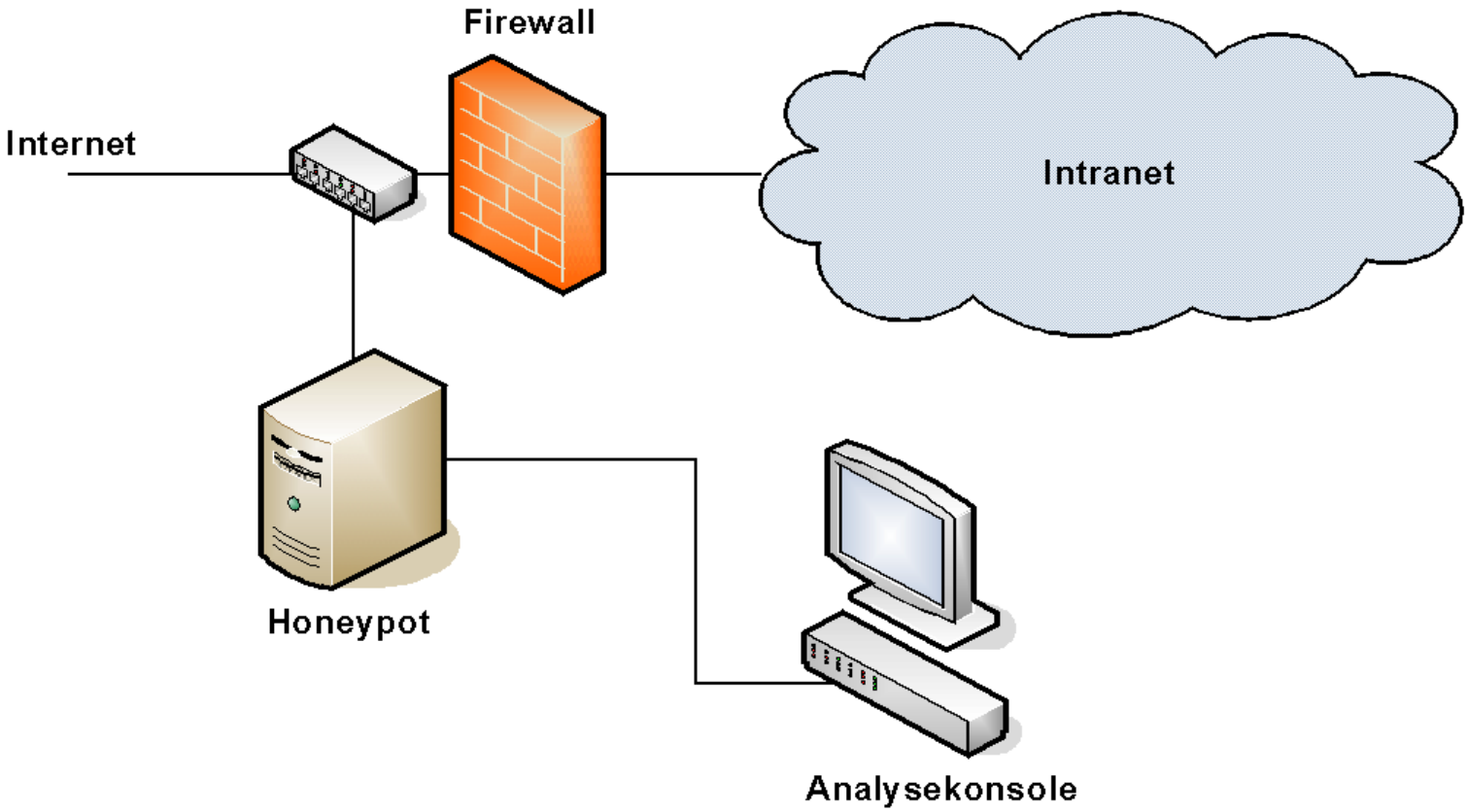
Motivation für den Einsatz von Honeypots

- Hacker in Fallen laufen zu lassen und dabei zu beobachten ist interessant und stellt einen besonderen Reiz dar.

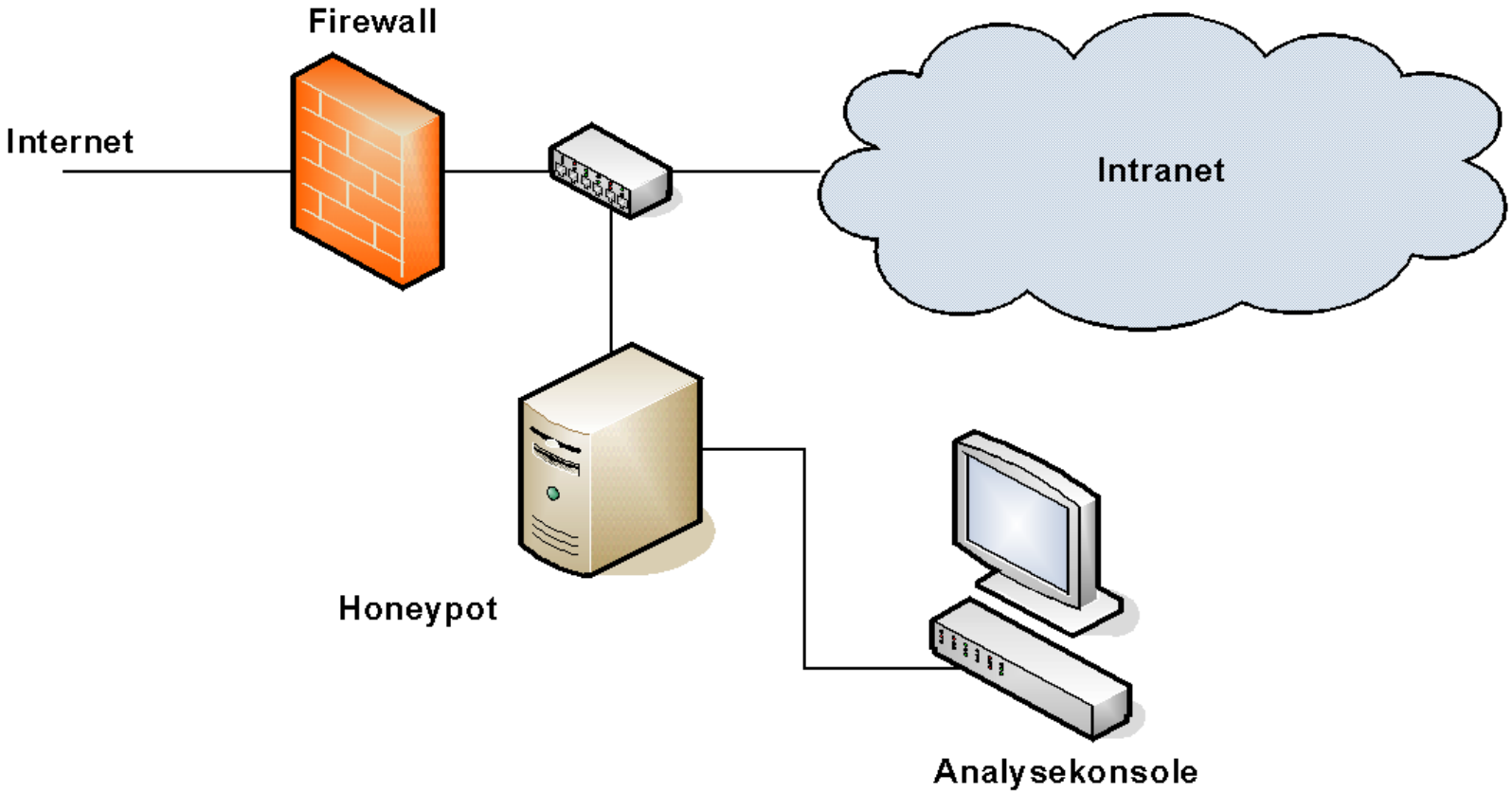
- „**Know Your Enemy**“
  - Technisches Lernen über Angriffsmethoden
  - Studie des Verhaltens von Angreifern
  - Entwicklung von Abwehrmaßnahmen und –strategien
  - Dokumentation und Veröffentlichung von Ergebnissen

- Integration in Sicherheitsarchitekturen
  - Schutz produktiver IT
  - Erfassung der Bedrohungslage

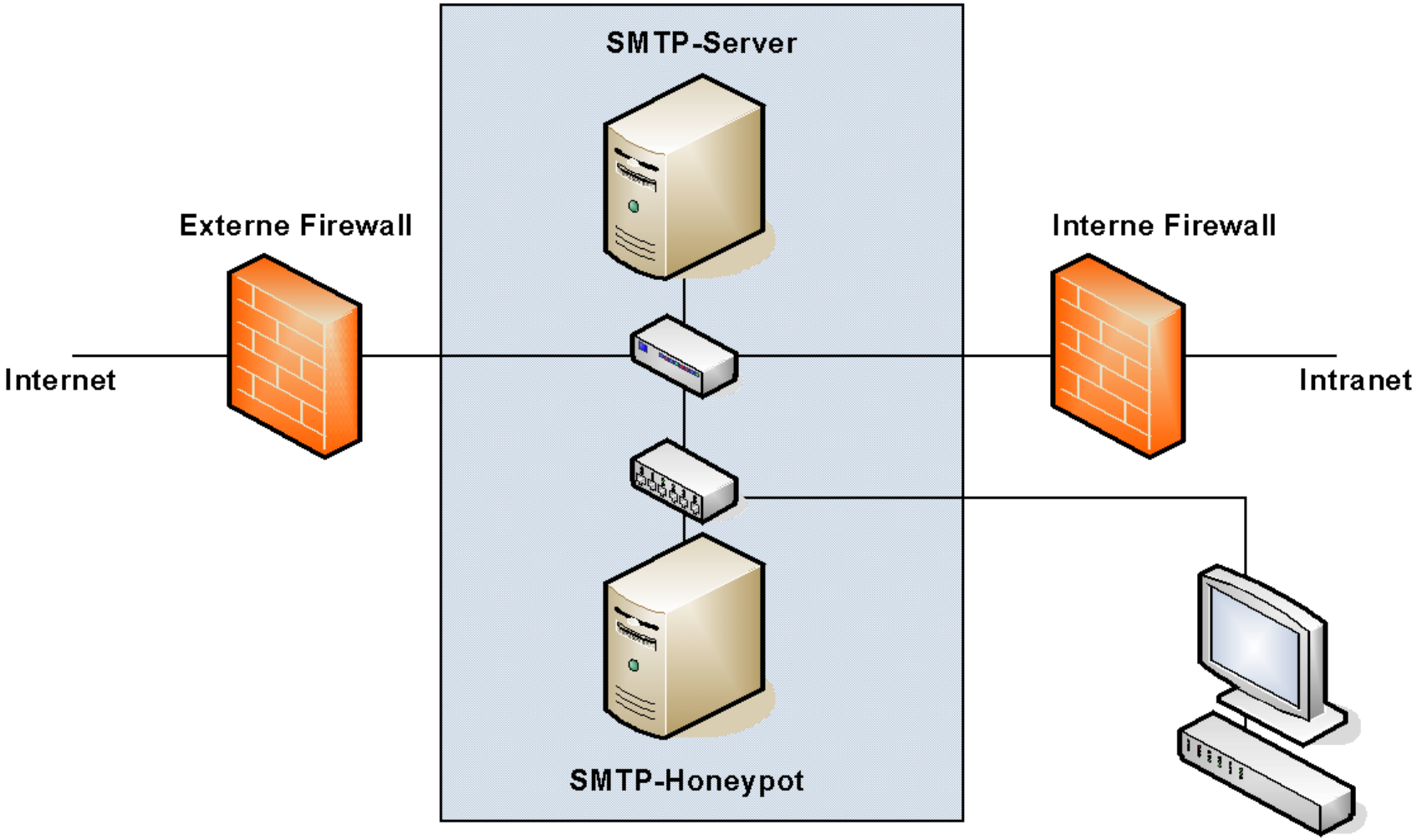
Honeypots in einer Sicherheitsinfrastruktur (1)



Honeypots in einer Sicherheitsinfrastruktur (2)



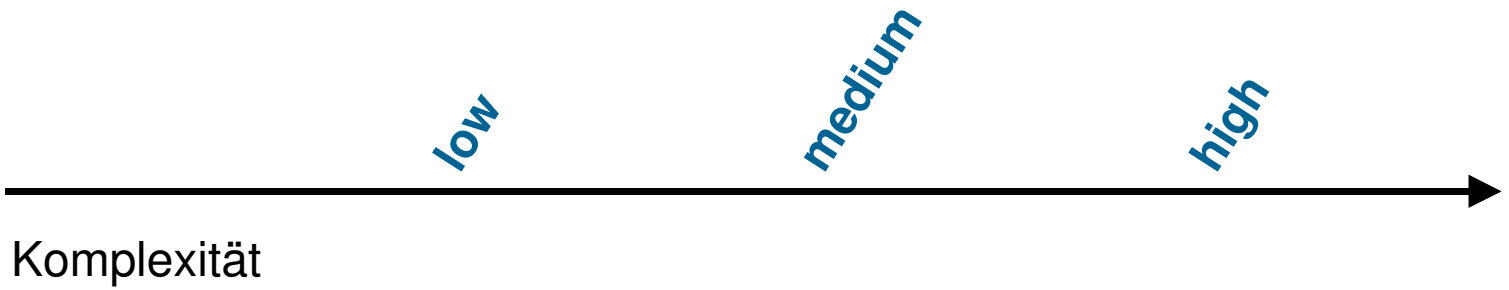
Honeypots in einer Sicherheitsinfrastruktur (3)



Kategorien von Honeypots

Bei Honeypots werden die folgenden Kategorien unterschieden:

low-interactive	keine Interaktion mit dem Angreifer Beispiel: Firewall-Logs
medium-interactive	beschränkte Interaktionsfähigkeit Beispiel: Emulation bekannter Schwachstellen
high-interactive	unbeschränkte Fähigkeit zur Interaktion Beispiel: reguläres System



Darknets

- Idee: **Pakete an nicht genutzte IP-Adressbereiche** sind suspekt.
- Im Vergleich mit Honeynets sind Darknets leicht zu implementieren. Es reicht aus, nicht genutzte IP-Adressen an ein **Sinkhole-System** zu routen, welches statistische Daten erfasst.
- Eine Auswertung des erfassten Traffics kann **Angriffstrends** wie eine Wurm-Ausbreitungen oder großflächige Scans offenbaren.
- Anhand von **Backscatter** lassen sich Rückschlüsse auf Angriffe mit gefälschten Absender-IP-Adressen aus dem Darknet-Bereich ziehen.
- Die **False-Positive-Rate** ist extrem gering.
- Allerdings kann aufgrund des **fehlenden Interaktionslevels** mit Darknets nur die erste Angriffsstufe erkannt werden.

honeyd

- Honeyd ist eine der ersten Honeypot-Entwicklungen. Seine Stärken liegen in der **Simulation unterschiedlicher TCP/IP-Stacks**.
- Dazu verwendet die Software nmap- und Xprobe2-Fingerprints diverser Betriebssysteme.
- Auf einem Host können bis zu 65535 virtuelle Systeme betrieben werden. Mit einem speziellen ARP-Daemon werden diese in eine bestehende Netz-Architektur integriert („Internet in a box“).
- Der Aufbau von honeyd ist modular und kann mit **Service-Skripten** um beliebige **Dienstsimulationen** erweitert werden. Für viele gängigen Dienste und Schwachstellen sind solche Skripte öffentlich verfügbar.
- Honeyd fällt in die Kategorie **low-interactive**.

Honeytokens

- Ein Honeytoken ist ein **Datum, das nicht regulär genutzt wird**. Alle Zugriffe sind suspekt und werden aufgezeichnet.

- Beispiele typischer Honeytokens:
  - Datenbank-Einträge
  - E-Mail-Adressen
  - DNS-Einträge (MX-Records)
  - IP-Adressen (vgl. Darknets)

- Stoll erstellte gefälschte SDI-Dokumente, mit denen er das Interesse des Hackers dauerhaft wecken wollte. Diese können ebenfalls als Honeytokens bezeichnet werden.

Nepenthes

- Nepenthes täuscht als medium-interactive honeypot einem Angreifer **Dienste mit bekannten Schwachstellen** vor.
- Angriffsversuche werden anhand von **Signaturen** erkannt. Nepenthes simuliert daraufhin ein verwundbares System, so dass weitere Angriffsstufen ablaufen können.
- Haupt-Einsatzgebiet ist derzeit das **Sammeln von Schadprogrammen**. Als Unix-Dienst ist die Software gegen die gängigen Windows-Attacken immun.
- Nepenthes kann nur auf **bekannte Angriffe** reagieren. Daten zu unbekannten Attacken werden nur eingeschränkt erfasst, bis ein entsprechendes **Verarbeitungsmodul** verfügbar ist.
- Nepenthes wird auch als **Komponente von Intrusion Detection Systemen** eingesetzt. Es dient dabei der Erkennung von Malware-Ausbreitungen in internen Netzen.

Clientseitige Honeypots

- Clientseitige Honeypots sind ein relativ neues Konzept. Sie warten nicht passiv auf Angriffsversuche sondern **bewegen sich aktiv im WWW**.

- Ziel ist es, ähnlich wie ein websurfender Benutzer bösartigen Webseiten zum Opfer zu fallen. Dazu wird das Web ausgehend von einem Einstiegspunkt systematisch gecrawlt.

- **Honeyclient** ist ein Set von Perl-Skripten. Diese verwenden den Internet Explorer mit zwischengeschaltetem Proxy für Inhaltsanalysen.

- Mit **gleichzeitiger Überwachung des Hosts** können Änderungen im Dateisystem und/oder in der Registry-Datenbank festgestellt und aufgezeichnet werden.

Honeynets – Überblick

Honeynets bestehen meist aus **high interactive Honeypots**, also aus dedizierter Hardware oder virtuellen Systemen.

Hacker sollen mit **realen Systemen in einem Netzverbund** angelockt und ihre Aktivitäten detailliert protokolliert werden.

Ein Angreifer soll **möglichst lange auf dem System beobachtet werden**. Er darf deshalb nicht erkennen, dass er sich auf einem Honeypot befindet.

Der Hacker soll mit dem gekaperten System möglichst **keine dritten Systeme angreifen** können, aber gleichzeitig Netzzugang haben, um zum Beispiel Programmcode nachzuladen, der für weitere Angriffsstufen benötigt wird.

Zentrale Herausforderungen in einem Honeynet sind **Data Capture**, also das Aufzeichnen von Aktivitäten, und **Data Control**, also die Kontrolle jeglicher Information.

Honeynets – The Honeynet Project

Zu Beginn stellten Sicherheitsexperten reguläre Systeme ins Netz mit dem Ziel, Hacker anzulocken und deren Aktivitäten aufzuzeichnen.

Gemeinsames Ziel: Angriffstechniken sollen erkannt und Gegenmaßnahmen dazu entwickelt werden.

Im Jahr 1999 formierte sich aus dieser Initiative das **Honeynet Project** unter dem Motto „**Know Your Enemy**“.

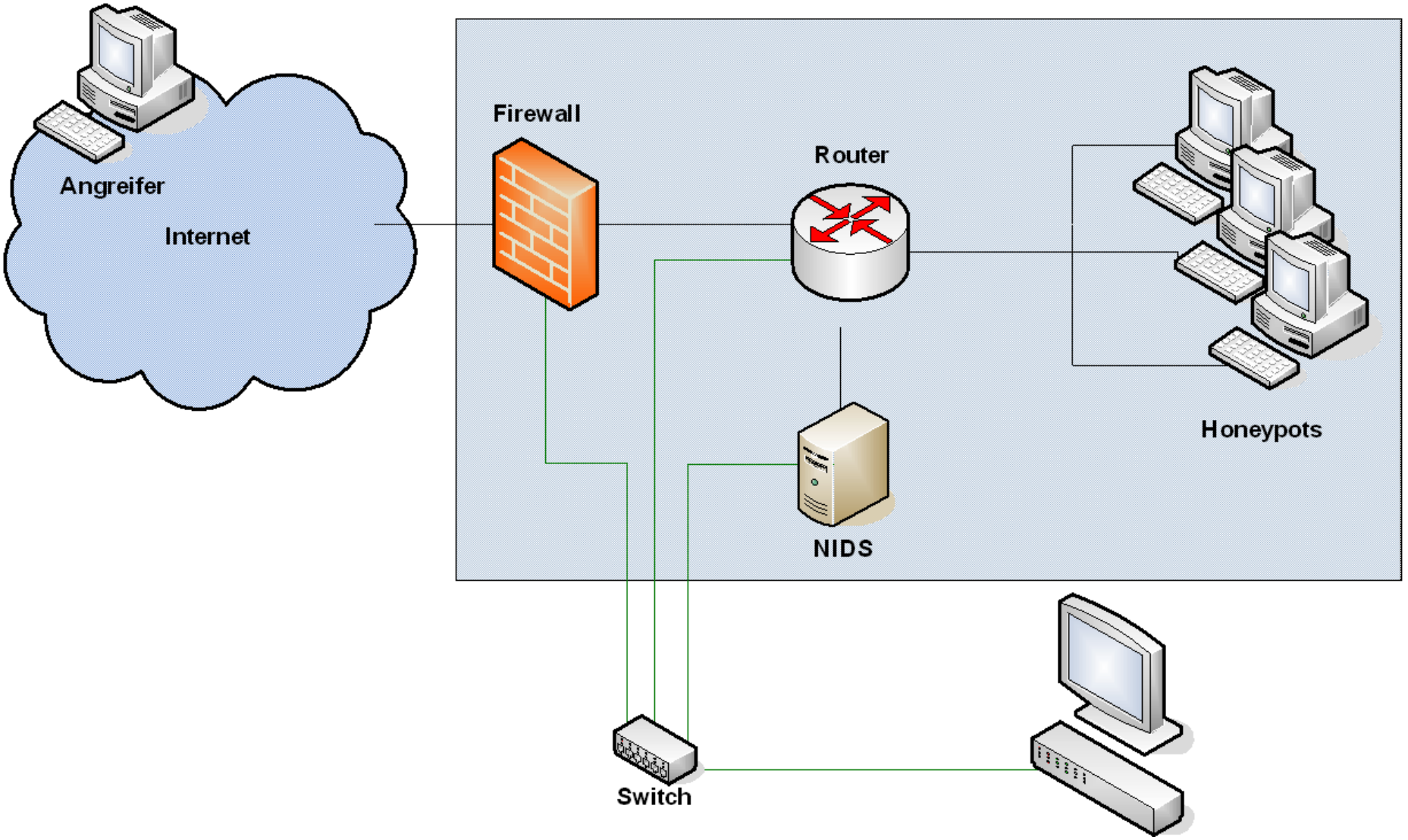
Die Roadmap des Projekts enthält vor allem die Entwicklung einer Honeynet-Architektur in drei Ausbaustufen: **GenI-**, **GenII-** und **GenIII-Honeynets**.

Als Mutterprojekt fördert das Honeynet Project Ablegerinitiativen wie das **German Honeynet Project**. Diese Ablegerprojekte müssen bestimmte Anforderungen erfüllen. Unter anderem muss ein **halbjährlicher Statusbericht** zur aktuellen Arbeit und Forschung veröffentlicht werden.

Gen I-Honeynets

- Honeynets der ersten Generation enthalten **Hosts mit unterschiedlichen Betriebssystemen** wie Microsoft Windows 2000 oder Linux.
- Die einzelnen Honeypots werden über **Netzkoppelelemente** wie Router und Switches miteinander verbunden und einem oder mehreren IP-Netzen zugeordnet.
- Ein **Network Intrusion Detection System** (NIDS) wie Snort wird an zentraler Stelle platziert, um Angriffe gegen die Honeypots zu erkennen.
- Eine **zentrale Firewall** beschränkt ein- und ausgehenden Netzverkehr auf Adress-, Protokoll- und Port-Basis. Data Control findet nur mit Hilfe der Firewall statt.
- Die zentralen Komponenten sind von einer **Analysekonsole** aus zugänglich, über welche die Auswertung der erfassten Daten erfolgt. Alle Informationen aus Data-Capture-Prozessen werden hier ausgewertet.

Beispieldesign für ein Gen I-Honeynet



Nachteile eines Gen I-Honeynet

- Honeynets der ersten Generation sind aufgrund der Restriktionen des Netztraffics **leicht zu erkennen** – Verbindungen werden großzügig angenommen, ausgehend aber spürbar beschränkt.
- Router und Firewall **dekrementieren das Time-To-Live-Feld** im IP-Header und sind somit nicht transparent.
- Router und Firewall sind aus dem Internet ansprechbar und potenziell verwundbar, können also angegriffen werden.
- Das Logging lokaler Aktivitäten findet mit üblichen Betriebsmitteln statt (Log-Dateien, netzbasiertes Logging wie Syslog). **Log-Informationen sind somit manipulierbar** – ein Angreifer kann so seine Spuren verwischen.
- **Verschlüsselte Kommunikation ist nicht einsehbar**, da nur auf Netzebene mitgelesen werden kann.

Gen II/Gen III-Honeynets

- Honeynets der zweiten Generation ergänzen GenI-Honeynets um ein spezielles Linux-basiertes Sicherheitsgateway, die **Honeywall**.
- Die Honeywall übernimmt alle Data-Control-Funktionen. Sie arbeitet als **Ethernet-Bridge** und ist damit sowohl für die Honeypots als auch für Hosts in externen Netzen **völlig transparent**.
- Die Filterung des Netz-Traffics und das Beschränken ausgehender Verbindungen erfolgt mittels **iptables**. Gleichzeitig können mit **Argus** Flow-Informationen erfasst werden.
- Das Intrusion-Prevention-System **snort inline** kann einige Angriffsversuche signaturbasiert erkennen und so modifizieren, dass sie ungefährlich werden.
- Lokale Aktivitäten auf den Honeypots werden mit dem **Logging-Tool Sebek** erfasst und an die Honeywall übertragen. Die **Analysekonsole** benötigt daher nur noch Zugriff auf die Honeywall. In der Regel wird dieser über ein dediziertes Netz bereitgestellt.

Sebek

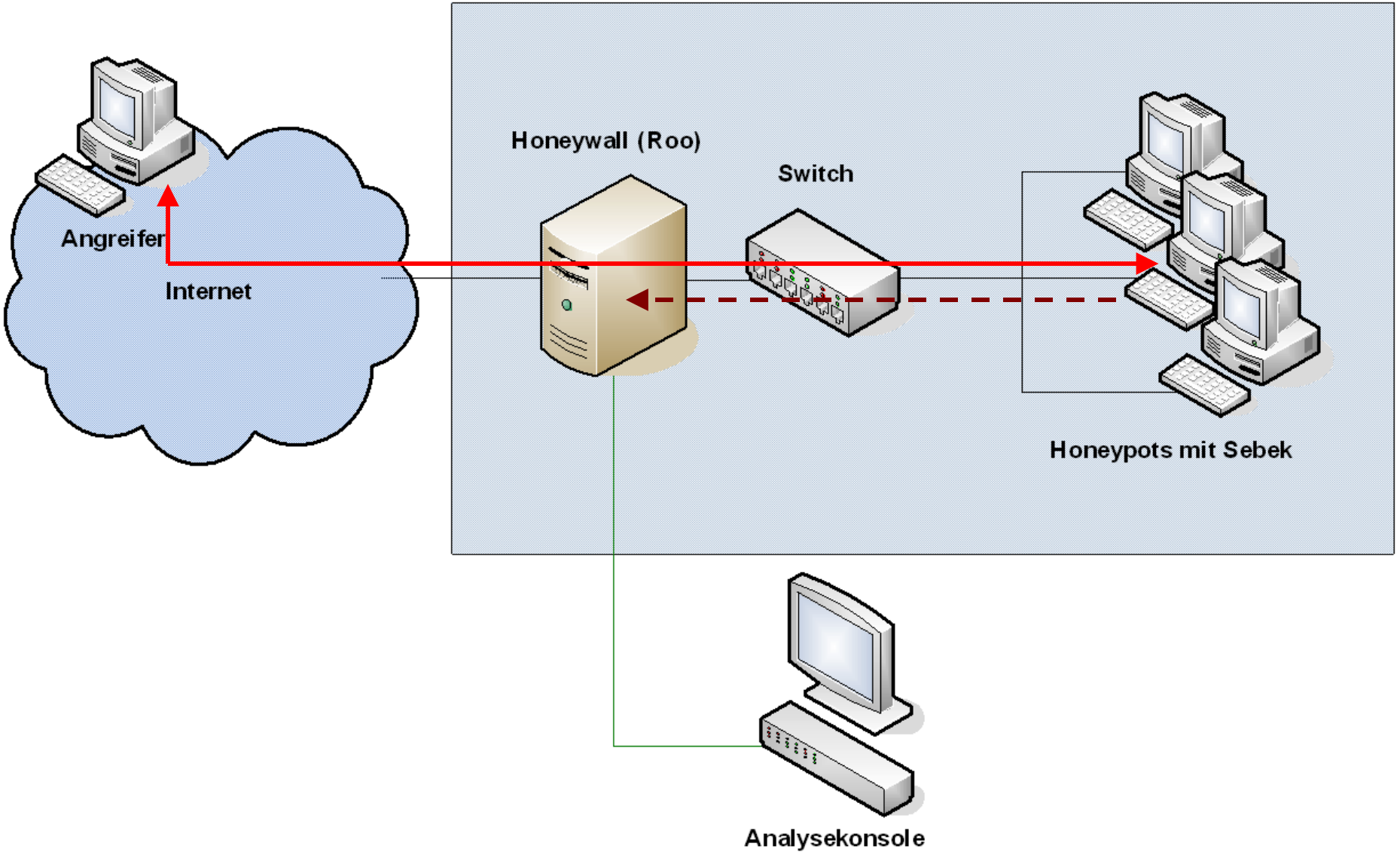
- Sebek ist das am weitesten entwickelte **Data-Capture-Tool**. Es basiert auf Rootkit-Technologie (System Call Table Hijacking) und ist für Windows-, Linux- und einige Unix-Systeme verfügbar.

- Das Kernel-Modul kann alle **Tastatur-Eingaben** sowie **Prozess-Starts** und **Dateioperationen** aufzeichnen. So lassen sich auch Daten aufzeichnen, die über verschlüsselte Verbindungen übertragen werden.

- Die Sebek-Daten werden per UDP an die Honeywall übermittelt. Auf dem Honeypot lassen sich die dazugehörigen Pakete und Prozesse nicht ohne weiteres detektieren.

- Das **Walleye-Webfrontend** der Roo-Firewall kann Sebek-Daten aufbereiten und übersichtlich darstellen. Netz-Traffic von anderen Verbindungen kann mit diesen Informationen korreliert werden.

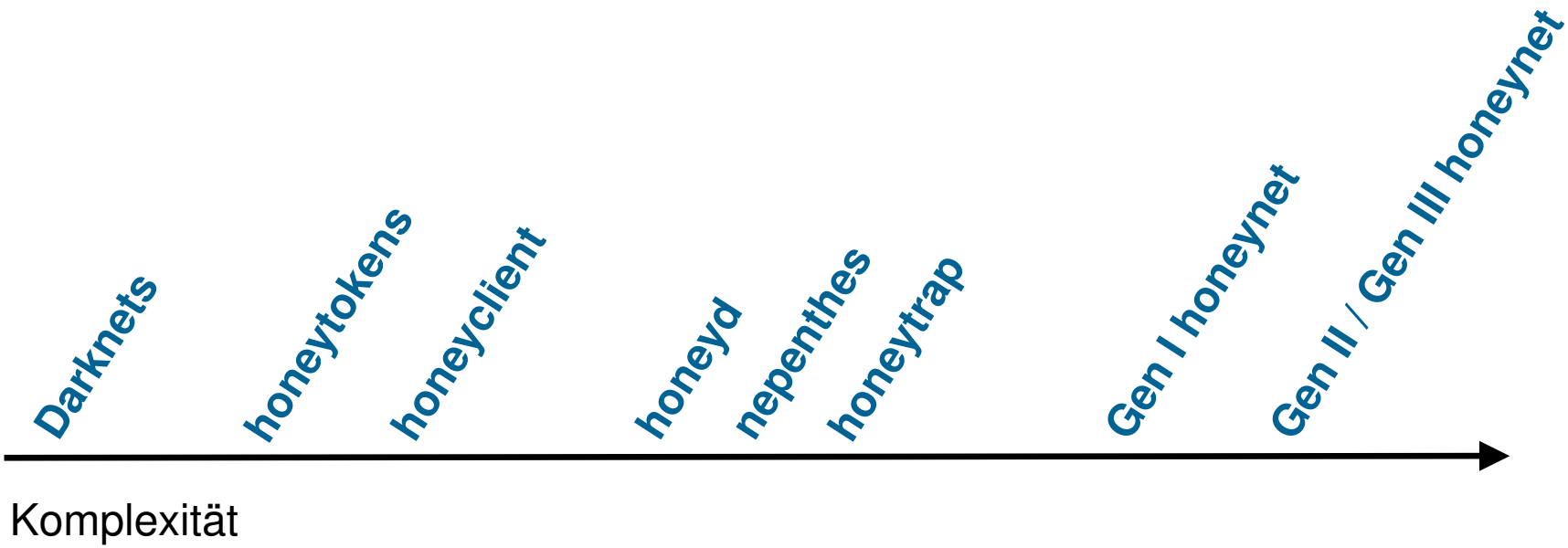
Beispieldesign für ein Gen III-Honeynet



Live-Demo:

Honeywall

Komplexität der vorgestellten Techniken



Die Komplexität wächst proportional mit steigendem Interaktionslevel.

Honeytrap – Überblick

Honeytrap ist ein **low-interactive honeypot** für POSIX-konforme Betriebssysteme.

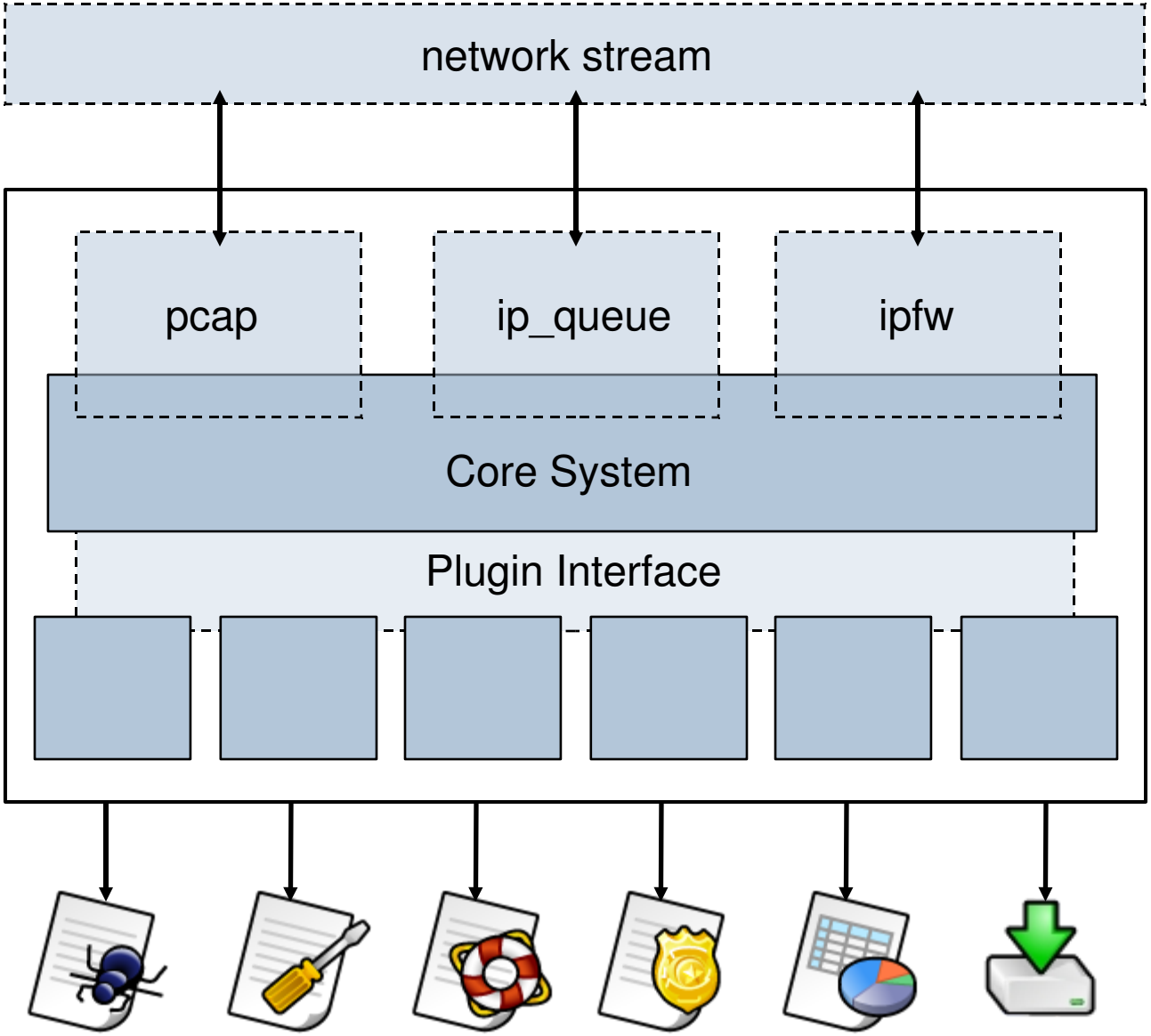
Zentrale Idee: **Angriffsversuche** werden detektiert und **dynamisch verarbeitet**, so dass Attacken aus Angreifersicht erfolgreich verlaufen.

Dadurch können auch völlig neue, bis dahin **unbekannte Angriffsmethoden sinnvoll behandelt** und wertvolle Informationen gewonnen werden, die beispielsweise für IT-Frühwarnung geeignet sind.

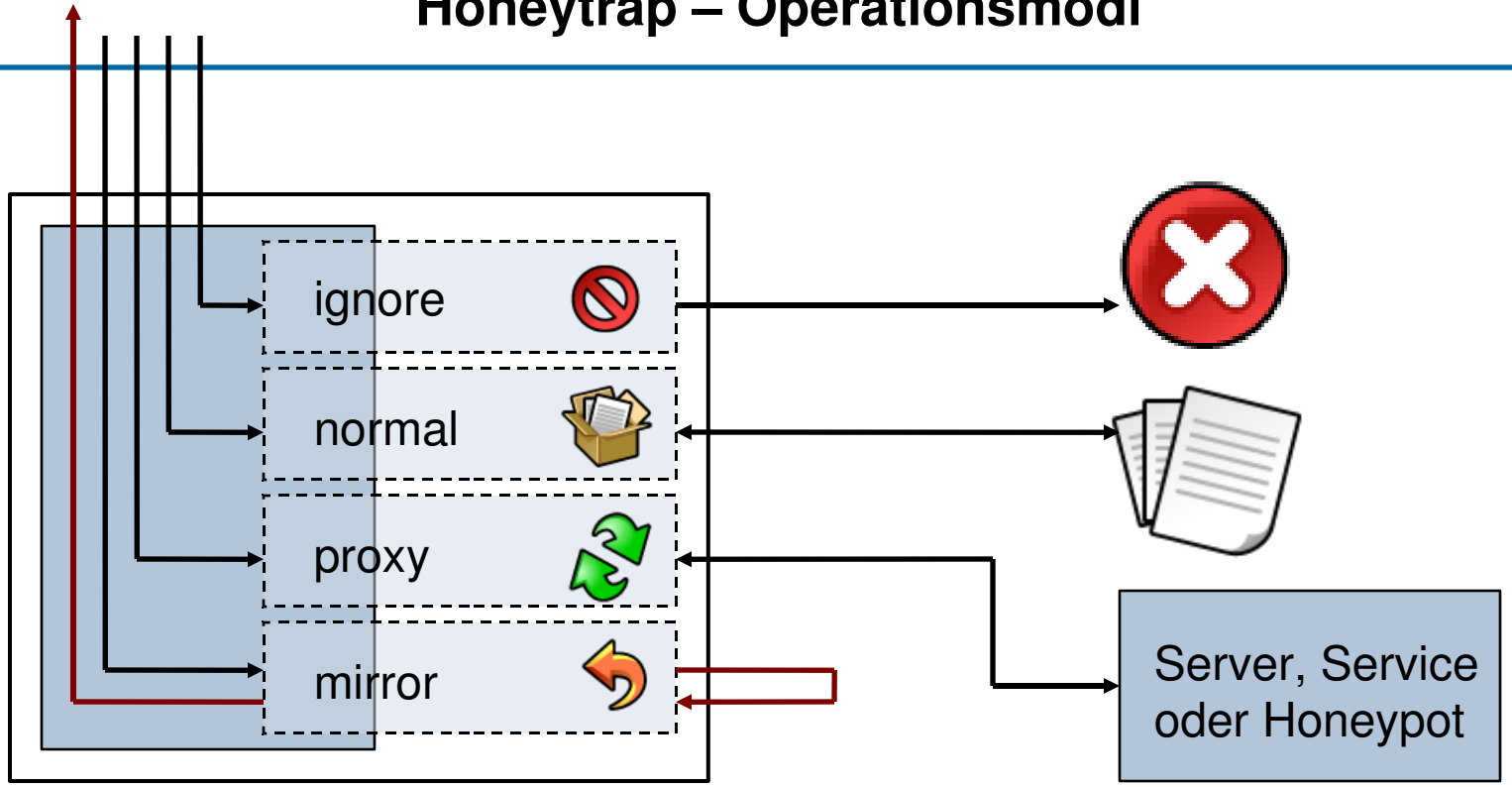
Datensammlung und Analysen werden strikt getrennt. Das Kernsystem stellt Capture-Mechanismen bereit, Angriffsanalysen werden mit Plugins teilautomatisiert oder vorbereitet.

Im **Mirror-Mode** nutzt honeytrap die Tatsache aus, dass die Mehrheit der Angriffe von seinerseits kompromittierten Systemen ausgeht, welche also dieselbe Schwachstelle besitzen. Gespiegelte Angriffe sind oft erfolgreich und die aufgezeichneten Daten daher von großem Wert.

Honeytrap – Software-Architektur



Honeytrap – Operationsmodi



Honeytrap unterscheider **vier Modi** zur Verarbeitung von Angriffsversuchen.
Der Operationsmodus kann **pro TCP- oder UDP-Port** definiert werden.

Honeytrap ist damit für den **Einsatz als Meta-Honeypot** prädestiniert.

Honeytrap – Erfasste Informationen

Informationen, die mit Honeytrap erfasst werden, umfassen vor allem:

- Start- und Endzeit eines Angriffs
- IP-Adressen des Angreifers und des Honeypots
- Verwendetes Protokoll
- Angesprochener TCP- oder UDP-Port des Honeypots
- Übertragene Nutzdaten (TCP- oder UDP-Payload)
- Länge der Nutzdaten in Bytes
- MD5-Checksumme der Nutzdaten

Live-Demo:

Honeytrap im Einsatz

Live-Demo:

Analyse einer UDF-Injection-Attacke

Sticky Honeypots (Tarpits)

- *Sticky Honeypots* (auch Tarpits) versuchen, **Ressourcen** eines Angreifers **möglichst lange zu binden**. Damit werden sie von Angriffen gegen weitere Systeme abgehalten.

- **Spam-Honeypots** verwenden Möglichkeiten des SMTP-Protokolls, um Verbindungen zu verlangsamen. Dies geschieht, indem SMTP-Statuscodes ein Minuszeichen vorangestellt wird. Der Client wird damit darüber unterrichtet, dass die Antwort des Servers noch unvollständig ist.

- Die Software **LaBrea** versucht, TCP-Verbindungen aufrecht zu erhalten, ohne dass Daten übertragen werden können.
  - Dazu übernimmt sie automatisch alle **nicht genutzten IP-Adressen** in einem Netzbereich und **beantwortet Verbindungsversuche** auf bekannten Ports.
  - Nach dem Dreiwege-Handshake werden regelmäßig ACK-Segmente mit sehr kleinem **window advertisement** übermittelt, so dass kein Timeout erreicht wird, aber Daten nur sehr langsam übertragen werden können (persistent mode).

Poisoned Honeypots

- *Poisoned Honeypots* sollen einem **Angreifer Schaden zufügen**.
- CIA-Bericht „*The Farewell Dossier – Dumping the Soviets*“: <https://www.cia.gov/csi/studies/96unclass/farewell.htm>
  - Während des Kalten Krieges setzten die USA entsprechende Methoden ein, um der russischen Industriespionage zu begegnen. Dabei wurden **speziell manipulierte Informationen** über das Design von Computerchips so platziert, dass sie vom KGB gefunden werden mussten.
  - Die verheerende **Explosion einer russischen Gaspipeline** wird auf den Einbau solcher fehlerhaften Chips in Turbinen zurückgeführt.
- Vergiftete Honeypot-Systeme enthalten Schadcode wie Viren, Würmer oder Trojaner, welcher das System des Angreifers befällt.
- Poisoned Honeypots sind grundsätzlich nur von theoretischem Interesse.

Identifikation von Honeypots

- **Virtuelle Maschinen** können anhand individueller Charakteristiken oft leicht erkannt werden. VMWare beispielsweise reagiert typisch auf bestimmte System Calls und kann außerdem anhand der Device-Namen erkannt werden. Sebek lässt sich ebenfalls detektieren.
- **Abweichungen im Verhalten der Dienste** deuten auf Manipulation hin.
- Bei betriebssystemspezifischen emulierten Diensten kann getestet werden, ob der **TCP/IP-Stack einen dazu passenden Fingerprint** besitzt.
- High-interactive Honeypots unterliegen in der Regel Beschränkungen in Bezug auf ausgehende Netz-Verbindungen.
  - Ein Angreifer kann Pakete, welche gängige IDS-Signaturen triggern, an einen weiteren, von ihm kontrollierten Host senden. Weist das empfangene Paket dann Unterschiede zum versandten auf, so ist dies ein **Hinweis auf ein Inline-IDS**.

VMWare-Identifikation in a Nutshell

```
#include <stdio.h>
int main () {
  unsigned char m[2+4], rpill[] = "\x0f\x01\x0d\x00\x00\x00\x00\xc3";
  *((unsigned*)&rpill[3]) = (unsigned)m;
  ((void(*)())&rpill)();

  printf ("idt base: %#x\n", *((unsigned*)&m[2]));
  if (m[5]>0xd0) printf ("Inside Matrix!\n", m[5]);
  else printf ("Not in Matrix.\n");
  return 0;
}
```

Quelle: RedPill <http://invisiblethings.org/papers/redpill.html>

Durch Ausführen der hexadezimal codierten „Store Interrupt Descriptor Table“-Anweisung wird das **Internet Descriptor Table Register** ausgelesen. Die **SIDT-Anweisung benötigt lediglich Ring3**-Privilegien und kann damit von VMWare nicht detektiert und abgefangen werden.

VMWare muss das IDTR des Gast-Systems in den Arbeitsspeicher mappen, so dass **keine konkurrierenden Zugriffe** mit dem Host-System auftreten können. Tests haben ergeben, dass VMWare das IDTR immer an eine Adresse oberhalb von 0xFF000000 mappt.

Quellen und Links

- *Honeypots – Tracking Hackers*, Lance Spitzner, 2003, Addison-Wesley
- *Know Your Enemy – Learning about Security Threats*, The Honeynet Project, 2004, Addison-Wesley

- The Honeynet Project, <http://www.honeynet.org>
- The German Honeynet Project, <http://pi1.informatik.uni-mannheim.de/projects/honeynet/overview>
- The Team Cymru Darknet Project, <http://www.cymru.com/Darknet/index.html>
- honeyd, <http://www.honeyd.org>
- nepenthes, <http://nepenthes.mwcollect.org>
- honeytrap, <http://honeytrap.sourceforge.net>
- Honeyclient, <http://www.honeyclient.org>
- Remote OS Detection via TCP/IP Fingerprinting, <http://insecure.org/nmap/osdetect/>
- „Know Your Enemy“ Whitepapers, <http://honeynet.org/papers/kye.html>
- RedPill, <http://invisiblethings.org/papers/redpill.html>

Kontakt

Bundesamt für Sicherheit in der
Informationstechnik (BSI)

Tillmann Werner, Edwin Reusch
Referat 121 – CERT-Bund
Godesberger Allee 185-189
53175 Bonn

Tel: +49 (0)1888 9582-5110
Fax: +49 (0)1888 9582-5427

tillmann.werner@bsi.bund.de
edwin.reusch@bsi.bund.de

http://www.bsi.bund.de
http://www.cert-bund.de