PARASITIC COMPUTING
Semesterarbeit Ethik

Version 1.1

Luzian Scherrer
Jürg Reusser

4. August 2002

Zusammenfassung

Das Prinzip „Parasitic Computing" wurde erstmals im August 2001 öffentlich erwähnt [BFJB01], als es Wissenschaftlern der Notre Dame Universität im US-Bundesstaat Indiana gelang, fremde Rechenkapazität ohne Wissen und Einwilligung derer Besitzer zur Lösung mathematischer Probleme zu nutzen. In dem vorliegenden Dokument soll nun die auf dieser Grundlagenarbeit aufbauende Diplomarbeit „Parasitic Computing" der Hochschule für Technik und Architektur Bern in ihren ethischen Aspekten genauer betrachtet werden. Wir wollen uns im folgenden Text den potentiellen ethischen Problemen parasitärer Rechenmethoden annehmen, indem wir diese systematisch aufzeigen und mögliche Lösungswege darstellen.

Inhaltsverzeichnis

1 Einleitung **3**
- 1.1 Was ist ein Parasit? . . . 3
- 1.2 Parasitismus . . . 3
- 1.3 Der Computer als Parasit . . . 4
- 1.4 Parasitismus in Computernetzwerken . . . 4
- 1.5 Rentabilität des Parasitic Computing . . . 5

2 Auswirkungen parasitärer Rechnenmethoden **5**
- 2.1 Potentielle Probleme . . . 5
  - 2.1.1 (Um)nutzung fremder Ressourcen . . . 5
  - 2.1.2 Beeinträchtigung des Wirtes . . . 5
- 2.2 Reichweite parasitärer Rechenmethoden . . . 6

3 Erweiterte Zusammenhänge **6**
- 3.1 Protokolle mit Sicherheitslücken? . . . 6
- 3.2 Abwägen zwischen Schaden und Nutzen . . . 6
- 3.3 Vorsorgliche Massnahmen . . . 7
- 3.4 Gesetzliche Grundlagen . . . 7
- 3.5 „Erlaubte" verteilte Berechnungen . . . 7
- 3.6 Komerzielle Nutzung . . . 8
- 3.7 Künstliche Intelligenz und parasitäre Methoden . . . 8

4 Lösungen **8**
- 4.1 Zwiespalt . . . 8
- 4.2 Zutrittskontrollen in Netzwerken . . . 9
- 4.3 Kontrollierte Nutzung parasitärer Methoden . . . 9
- 4.4 Überwachung der Rechner im Netzwerk . . . 10
- 4.5 Abschalten von nicht essentiellen Services . . . 10

5 Anhang **11**
- 5.1 Abkürzungsverzeichnis . . . 11

1 Einleitung

1.1 Was ist ein Parasit?

Das Wort ,,Parasit" findet seinen geschichtlichen Ursprung im antiken Griechenland. Damals trugen gewisse Volksvertreter auf bestimmte Zeit den Status des ,,Parasiten" und waren als solche geheissen, die Auswahl des Getreides, des Brotes und der Speise für das kutlische Opfermahl zu treffen und dieses mit dem Priester einzunehmen. Darin erklärt sich auch gleich die Herkunft des Wortes: Neben-, mit- oder bei- (pará) speisender (sitós). Lateinisch bedeutet ,,parasitus" Tischgenosse oder eben, wie bekannt, Schmarotzer. Ursprünglich trug das Wort, ganz im Gegensatz zu heute, keinen negativen Beigeschmack; die Parasiten im antiken Griechenland waren hochgeachtete, religiöse Beamte.

Parasitentum wurde früher auch als Lebenskunst betrachtet. Lukian von Samosata [1] hielt vor knapp 2000 Jahren fest: ,, Alle anderen Künste sind ohne gewisse Werkzeuge (die mit Kosten angeschafft werden müssen) ihrem Besitzer unnütz; niemand kann ohne Flöte flöten, ohne Violine geigen, oder ohne ein Pferd reiten: einzig die Parasitenkunst ist sich selber so genug und macht es ihrem Meister so bequem. Andere Kunstverwandte arbeiten nicht nur mit Mühe und Schweiss, sondern grösstentheils sogar sitzend oder stehen, und zeigen dadurch, dass sie gleichsam Sclaven ihrer Kunst sind: der Parasit hingegen treibt die seinige auf eben die Art wie Könige Audienz geben, – liegend" (vgl. [Enz01]).

Im Laufe der Jahrhunderte änderte sich diese Ansicht jedoch, und Parasit verkam zum Schimpfwort. War beim antiken ,,Parásitós" noch ganz klar, dass es sich dabei um eine Person aus Fleisch und Blut handelte, driftete man mit der Bezeichnung zunächst weit in die Botanik, beispielsweise zu Misteln, Moosen und Flechten – den heutigen Phytoparasiten, dann in in die niedere Tierwelt zu Läusen, Bandwürmern und Bazillen – heute Zooparasiten, ab.

1.2 Parasitismus

Die Ernährungsbeziehung zwischen dem Parasiten und seinem Wirt trägt die Bezeichnung Parasitismus. Der Parasit kann den Wirt in Einzahl (Solitärparasitismus) oder in Mehrzahl (Gregärparasitismus) befallen. Zu starker Gregärparasitismus (Superparasitismus) schadet dem Parasiten, weil er sich selbst die Lebensgrundlage entzieht. Wenn verschiedene Parasitenarten gleichzeitig in einem Wirt schmarotzen (Multiparasitismus), bleibt meist nur einer am Leben, oder alle gehen zugrunde. Parasiten können auch selbst wieder von Parasiten befallen werden (Hyperparasitismus). Sonderfälle von Parasitismus sind Cleptoparasitismus und Staatsparasitismus; im weiteren Sinne auch Brutparasitismus und Nahrungsparasitismus.

Der Parasitismus stellt gewissermassen eine Alternative zur klassischen Räuber-Beute-Beziehung dar. Er ist eine Symbiose, bei der ein Organismus indirekten Nutzen aus einem anderen zieht. Genauergenommen handelt es sich beim Parasitismus um eine kommensalistische, also eine einseitige Symbiose, bei welcher im Gegensatz zur mutualistischen Variante nur einer der Beteiligten vom anderen profitiert und kein gegenseitiger Austausch besteht. Entscheidendes Merkmal parasitärer Beziehungen ist immer, dass der Parasit seinem Wirt naturgemäss mit Schonung begegnet (oder begenen

[1] griechischer Schriftsteller, geb. in Samosata am Euphrat um 120 n. Chr., gest. nach 180.

muss), da er von diesem entscheidend abhängt, ja in seiner Existenzgrundlage auf ihn angewiesen ist.

1.3 Der Computer als Parasit

Der parasitäre Computer (vgl. [BFJB01]) nun, oder etwas genauer ausgedrückt das parasitäre Rechenwerk (welches nur einen Teil eines Computers im eigentlichen Sinne darstellt) kann am ehesten mit dem Solitärparasitismus verglichen werden. Es besteht aus einem einzelnen Knoten im Netzwerk (Parasit), das beliebige andere Knoten (Wirte) für sich arbeiten lässt. Das von Forschern der amerikanischen Notre Dame University definierte Verfahren (vgl. [Fre02]), welches hierzu benutzt wird, soll nun zum weiteren Verständnis im folgenden Abschnitt seinem Prinzip nach kurz erläutert werden.

1.4 Parasitismus in Computernetzwerken

Die Kommunikation innerhalb von Computernetzwerken wie beispielsweise dem Internet, basiert aus Gründen der Kompatibilität zwischen unzähligen verschiedenen Hardwareherstellern auf wohldefinierten, durch internationale Gremien beglaubigten und veröffentlichten Standards (vgl. [For02]). Bei diesen Standards handelt es sich um sogenannte Kommunikationsprotokolle, welche exakt vorgeben, in welcher Form ein Computer im Netzwerk Nachrichten zu verschicken, und wie er sich beim Empfang von Nachrichten zu verhalten hat. Solche Nachrichten sind Datenpakete, welche einerseits aus den frei definierbaren Nutzdaten, andererseit aus den sich aus eben diesen Protokollstandards ergebenden Kontrolldaten bestehen.

Einen Teil der Kontrolldaten bildet bei den weitverbreitetsten Kommunikationsprotokollen (namentlich IP, UDP, TCP, ICMP – dies sind die dem gesamten Internet zugrundeliegenden Standards [Ste94]) die sogenannte Internet-Checksumme. Die Checksumme dient der Überprüfung von Datenpaketen auf ihre Korrektheit. Der Absender bildet also über die Nutzdaten seiner zu sendenden Nachricht eine nach einem bestimmten Algorithmus berechnete Prüfsumme (Checksumme) und schickt diese zusammen mit der eigentlichen Nachricht an den Empfänger. Dieser ist nun vom Protokollstandard aufgefordert, zuerst die Checksumme nachzurechnen, und erst wenn diese mit den Nutzdaten übereinstimmt, je nach Anwendung mit der Verarbeitung der restlichen Daten weiterzufahren. Stimmt die Checksumme nicht mit den gesendeten Nutzdaten überein, so muss das Datenpacket stillschweigend verworfen werden.

Bei der Checksummenprüfung auf Empfängerseite kommt das Prinzip des „Parasitic Computing" zum tragen, durch welches eine Kandidatlösung eines durch den Checksummenalgorithmus abbildbaren Problemes vom Emfpänger auf Korrektheit überprüft werden lassen kann. Wird das Datenpacket vom Empfänger weiterverarbietet (dies lässt sich Aufgrund dessen weiteren Verhaltens mit den Nutzdaten bestimmen), so war die zu prüfende Kandidatlösung korrekt, andernfalls wird sie vom Empfänger verworfen und war nicht korrekt. Konkret lassen sich mit diesem Prinzip die binären Grundoperationen XOR und AND ausführen, und auf dieser Grundlage widerum sind sämtliche Operationen eines klassischen Computers aufbaubar (siehe auch [Wir95]). Es lässt sich auf diese Weise also ein parasitäres Rechenwerk realisieren, welches in seinem Kern ausschliessich auf Fremdressourcen aufgebaut ist.

Dabei ist von entscheidender Bedeutung, dass die vom Empfänger – also dem Wirt – genutzten Ressourcen von diesem willentlich zur Verfügung gestellt und benutzt werden dürfen, allerdings durch den Parasiten anders als ursprünglich vorgesehen verwendet werden.

1.5 Rentabilität des Parasitic Computing

An dieser Stelle muss noch erwähnt werden, dass das ,,Parasitic Computing" in seiner momentan bekannten Form nicht rentabel einsetzbar ist. Der Aufwand, Datenpackete für die beschriebene Umnutzung zu generieren, zu versenden und die Ergebnisse zu prüfen, übersteigt den daraus zu gewinnenden Nutzen um ein Vielfaches. ,,Parasitic Computing" ist daher in seiner aktuellen Variante nur ein ,,Proof of Concept", womit allerdings nicht festgelegt ist, dass nicht in Zukunft gewinnbringende Anwendungen gefunden werden könnten. Die potentiellen Ausnutzungsmöglichkeiten sind sehr weitreichend und eine Vielzahl an Protokollen, interessant dürften beispielsweise solche kryptographischer Natur sein, könnten auf ähnliche Weise Rechner zu Wirten von Parasitismus machen.

2 Auswirkungen parasitärer Rechnenmethoden

2.1 Potentielle Probleme

In den folgenden Abschnitten sollen die sich aus der parasitären Berechnung mittels Fremdressourcen ergebenden Problematiken genauer aufgezeigt werden.

2.1.1 (Um)nutzung fremder Ressourcen

Wie bereits in Abschnitt 1.4 angetönt, basiert das Parasitic Computing auf Mechanismen, die der Öffentlichkeit vom Wirt zur Verfügung gestellt und so prinzipiell durch jedermann – implizit erlaubt – benutzt werden dürfen. Dies tut der Parasit allerdings nicht zur vorgesehenen Anwendung der Fehlerprüfung der Kommunikation mit dem Wirt, sondern ausschliesslich für seine eigenen, dem Wirt absolut unnützen Zwecke. Der Wirt verliert also den seinen Nutzen an den Ressourcen die er verfügbar macht, nämlich der von ihm naturgemäss gewollten fehlerfreien Kommunikation mit anderen Rechnern.

Es drängt sich demnach die Frage auf, ob nun effektiv noch die ursprünglich zur Verfügung gestellten Ressourcen benutzt werden, oder eher, bedingt durch die Umnutzung des Parasiten, unwissentlich ausnutzbare Ressourcen beansprucht werden.

2.1.2 Beeinträchtigung des Wirtes

Nun ist es natürlich entscheidend zu wissen, in welchem Masse der Wirt diesen Parasitismus zu spühren bekommt, oder etwas expliziter ausgedrückt, in welchem Ausmass er dadurch beeinträchtigt wird.

Auf der untersten Ebene der durch den Parasiten ausgelösten Operationen liegen zwei atomare Grundfunktionen (siehe Abschnitt 1.4). Jede parasitäre Einheit besteht aus ausschliesslich einem dieser Atome, und erst wenn ein Wirt eine ganze Serie – es sind

dies selbst für primitive Operationen wie etwa die Multiplikation zweier Zahlen bereits mehrere tausende – solcher Einheiten ,,bewirtet", lässt sich so für den Parasiten etwas sinnvolles erreichen. Eine einzelne solche Einheit fällt für den Wirt kaum ins Gewicht, kann also aus praktischer Sicht nicht, wohl aber aus ethisch-theoretischer Sicht problematisch sein.

Das dem Solitärparasitismus entsprechende ,,Parasitic Computing" bedient sich nun allerdings nicht eines einzigen Wirtes, sondern verteilt die Arbeit auf eine Unmenge von Rechnern, im Idealfall so, dass pro Wirt nur eine der atomaren Grundoperationen ausgeführt wird. Ein einzelner Parasit wird in der Praxis also kaum feststellbaren Schaden anrichten können.

2.2 Reichweite parasitärer Rechenmethoden

Auf die Nichtrentabilität der Methode wurde bereits in Abschnitt 1.5 eingegangen, und daraus folgt, dass ,,Parasitic Computing" zum momentanen Zeitpunkt nur für Experimente eingesetzt werden kann.

Denkbar ist aber natürlich auch, dass Applikationen entwickelt werden, welche bösartig Datenpakete nach dem parasitären Prinzip verteilen, mit dem Ziel, einen fremden Rechner zu überlasten und diesen somit auszuschalten. Solche Missbräuche sind in Computernetzwerken weitreichend bekannt unter dem Namen ,,denial of service attack" (vgl. [Fer00]).

3 Erweiterte Zusammenhänge

Nachfolgend einige Punkte, welche erweiterte Zusammenhänge und Erklärungen zum Thema "Parasitic Computing" und Ethik liefern sollen.

3.1 Protokolle mit Sicherheitslücken?

Mittlerweilen ist es bekannt und öffentlich, dass zum Beispiel mittels ICMP oder TCP primitive Grundoperationen (XOR und AND) berechnet werden können. Naheliegenderweise sollten mit dieser Erkenntnis derartige Protokolle für unsicher, oder zumindest problematisch erklärt deshalb nicht mehr verwendet werden. Dies ist aber nicht praktisch durchführbar, denn solche weitverbreiteten, etablierten Kommunikationsmechanismen können unmöglich kurzerhand für ungültig erklärt oder ausgeschaltet werden. Das halbe Internet beispielsweise basiert auf TCP. Falls nun TCP nicht mehr verwendet werden sollte, würde das Internet nicht mehr in diesem Sinne existieren können. Eine Änderung oder Anpassung in den Kommunikationsstandards würde weltweit enorme Auswirkungen mit sich bringen und auf Hersteller- wie auch auf Anwenderseite auf heftigen Widerstand stossen.

3.2 Abwägen zwischen Schaden und Nutzen

Erwiesen ist, dass Resourcen, wie etwa Berechnungen mittels ICMP oder TCP, nicht rentabel sind. Das bedeutet, dass zum zusammenstellen eines Paketes, dass eine einzige atomare Grundoperation (binäres XOR oder AND) verteilt berechnen kann, ein Vielfaches an Rechenleistung erforderlich ist. Zudem stellt die Internetverbindung in

jedem Fall einen Flaschenhals dar, denn um Rechenkapazitäten zu erreichen wie diese etwa von einem modernen Prozessor zur Verfügung gestellt werden, würden auch die momentan schnellsten gebräuchlichen lokalen Netze mit Übertragungsraten von bis zu 1'000 MBit in der Sekunde bei Weitem nicht ausreichen. Kurz gesagt, die atomaren Grundoperationen der momentan bekannten Methode geben (noch) zu wenig her.

3.3 Vorsorgliche Massnahmen

Bereits haben einige Computerfirmen wie beispielsweise Microsoft oder Oracle, welche vielbesuchte und weltbekannte Internetseiten anbieten, die verzichtbareren Protokolle wie ICMP abgestellt (dies lässt sich mittels dem auf üblichen Bestriebssystemen unter dem Namen `ping` bekannten Befehl nachvollziehen). Das Protokoll ICMP dient primär zu Diagnose- und Kontrollzwecken und ist für den Anbieter einer Website nicht zwingend zu implementieren.

Solches Handeln hat natürlich verschiedene Gründe (siehe auch [Lut01]), einerseits versucht man auf diese Weise sämtlichen nicht zwingend nötigen Datenverkehr zu unterbinden, andererseits will man sich damit vor möglicherweise zukünftig bekanntwerdenden Sicherheitslücken schützen. Dies rein nach dem Prinzip, je kleiner die Möglichkeit, desto kleiner auch die Wahrscheinlichkeit. Fast täglich entdeckt die Internetgemeinde neue Sicherheitsrisiken in allen Arten von Programmen, Protokollen und Algorithmen, welche auf einschlägigen Websites – sei dies der Warnung oder der Bekanntmachung aus negativen Gründen wegen – publiziert werden (vgl. [(CE02]).

3.4 Gesetzliche Grundlagen

Wie nirgendswo anders als im diesbezüglich blutjungen Informatikbereich hinkt die Gesetzgebung den stetig und rasant wachsenden Technologieneuerungen hinten nach. Dies äussert sich primär dann auf fatale Weise, wenn in einem Land mit schlecht an neue Technologiemöglichkeiten angepasser Gesetzgebung zum Beispiel ein Virus entwickelt wird, welches landesgrenzenübergreifend Schäden in Milliardenhöhe anrichtet, der Täter aber durch das Netz der Gesetze fällt. Bestes Exempel dafür ist das erst kürzlich in den Medien ausgiebig behandelte Virus „I Love You" (siehe [Oeb00]).

Oben genanntes Beispiel des Entwicklers *Onel de Guzman* zeigt die typische Problematik auf, wie Gesetzgebungen, die in ihrem Sinne abschreckende Wirkungen haben sollten auf ethische Zuwiederhandlungen, im Bereich Informatik versagen und so einschlägige Entwickler und Hacker geradezu anspornen, EDV-Anlagen lahmzulegen, weil ja in den meisten Fällen viel Ruhm und Medienpräsenz und auf sie zukommt, meist jedoch ohne gesetzliche Konsequenzen, falls den Fehlbaren überhaupt faktisch ein Verbrechen nachgewiesen werden kann.

3.5 „Erlaubte" verteilte Berechnungen

Das zur Zeit populärste Projekt, welches fremde Rechenkapazitäten im professionellen Sinne legal nutzt, ist zweifelsohne das Unternehmen SETI (The Search for Extraterrestrial Intelligence [SET01]) der amerikanischen Berkeley Universität. Beim Projekt SETI gilt es, Unmengen von durch Radarstationen aus dem Weltall aufgezeichneten

Daten nach bestimmten Mustern abzusuchen. Dabei handelt es sich um riesige Datenmengen, die einzelner Computer mit heutiger technologie unmöglich bewältigen kann – es muss die Aufgabe also auf tausende von Maschinen verteilt werden.

Eine Applikation, welche von den Teilnehmenden auf deren Rechnern (freiwillig) installiert wird, lädt hierbei jeweils ein Datenpaket mit vorgängig aufgezeichneten Signalen zur lokalen Auswertung herunter. Diese Signale werden mit kleinster Priorität immer dann schrittweise verarbeitet, wenn der eigene Computer über freie Rechenresourcen verfügt und nicht anderweitig genutzt wird. Sobald ein Paket fertig ausgewertet ist, schickt die Applikation dieses wiederum via Internet zum Server zurück, welcher die Applikation mit einem ächsten Paket zur Auswertung versorgt, und so weiter.

3.6 Komerzielle Nutzung

Das Nutzen fremder Rechenkapazität bietet auch einen aus komerzieller Sicht interessanten Ansatzpunkt. So könnte beispielweise freie Rechenkapazität – und diese übersteigt die produktive Phase des Rechenwerkes auf den meisten Maschinen um ein vielfaches – gegen eine Entgeltung angeboten werden. Denkbar wäre, dass Privatpersonen Rechenzeit gegen Bezahlung auf Stundenbasis an Firmen verkaufen, was Letzteren die Möglichkeit eröffnet, massive verteilte Rechenwerke ohne Wartungs- und Betreibungsaufwand aufzubauen.

3.7 Künstliche Intelligenz und parasitäre Methoden

Momentan vielleicht grossenteils noch Science Ficiton, möglicherweise aber bald schon in der Realität im Einsatz sind Programme mit echter sogenannter künstlicher Intelligenz. Wie lange es dauern wird, bis sich solche Programme dann, so wie dies die Forscher von heute tun, durch eigene Methoden fremde Rechenkapazität verschaffen, wird sich wohl oder übel zeigen. Das Verhalten solcher Programme wird mit ihrem weiteren Fortschritt schwerer und schwerer Vorhersagbar und allfällige Auswirkungen sind kaum abzuschätzen.

4 Lösungen

Nachfolgend einige Punkte, welche aufzeigen sollen, dass sich die Suche nach Lösungen, parasitäre Rechenkapazitäten nicht zu ermöglichen respektive Missbräuche zu verhindern, alles andere als einfach gestaltet.

Folgende Überlegungen sind also nicht zwingendermassen als pfannenfertige Lösungsansätze zu verstehen, sondern auch als Denk- und Wegfindungsilfe, wie mit einem Zwiespalt zwischen Gutmütigkeit einerseits (Bereitstellung eines Service, beispielsweise ICMP) und einseitiger Ausnutzung („Parasitic Computing”) umgegangen werden kann.

4.1 Zwiespalt

Das ganze Prinzip der Vernetzung von Computern und Maschinen beruht darauf, irgendwelche Datenpakete, welche hier nicht näher erläutert werden sollen, sicher, performant und vor allem fehlerfrei durch ein Netzwerk von Knoten zu transportieren. Dazu müssen die beteiligten Rechner zwingendermassen Mechanismen und Services zur

Verfügung stellen, welche Möglichkeiten bieten, fehlerhafte Datenpakete zu identifizieren und entsprechend darauf zu reagieren. Nur so kann ein korrekter Datentransport gewährleistet werden.

Parasitäre Methoden der Fremdkapazitätsnutzung zielen genau darauf ab, solche Mechanismen der Transportkontrolle von oben erwähnten Datenpaketen zu missbrauchen.

Offensichtlicherweise können Services und Mechanismen, welche die unabdingbare Voraussetzungen bilden zur konsistenten Kommunikation innerhalb eines Netzwerkes, nicht abgeschaltet werden, da ja sonst der korrekte Transport von Datenpaketen nicht mehr gewährleistet wäre. Vielmehr müssen allgemeinverträgliche Kompromisse gefunden und realisiert werden, welche eine gute Balance zwischen Einschränkungen und Funktionalität bilden.

4.2 Zutrittskontrollen in Netzwerken

Ein Rechner, der an ein Netzwerk angeschlossen wird, soll so konfigueriert sein, dass auf ihm keine den andern Rechnern schädlichen Programme zum laufen gebracht werden können.

Ein gutes Beispiel für einen kontrollierten Zugang in ein LAN sind Computer, deren MAC-Adressen (hardwareabhängige, weltweit eindeutige Adresse pro Maschine) vorgängig beim zuständigen Verbindungsglied registriert werden müssen, bevor der Rechner überhaupt auf dem Netzwerk kommunizieren kann. Realisiert so beispielsweise in der Postfinance in Bern.

Fazit: Dieser Ansatz ist insofern lösungsrelevant, weil er eine Möglichkeit aufzeigt, wie unter normalen Umständen sicher verhindert werden kann, dass sich fehlbare Maschinen bösartigerweise in einem Netzwerk betätigen können.

4.3 Kontrollierte Nutzung parasitärer Methoden

Falls parasitäre Methoden innerhalb eines Netzwerkes angewendet werden, dann sollen diese Varianten der Rechenkapazitätsnutzung bewusst, kontrolliert und ohne schädliche Nebeneffekte vor sich gehen. Unerwünschte Effekte bilden dabei folgende möglichen Punkte:

- Überlastung des Netzwerkes
- Ungleiche Ausnutzung freier Rechenresourcen

Fazit: Dieser Ansatz soll verdeutlichen, dass freie Resourcen *sinnvoll* genutzt werden sollten. Damit kann aber keineswegs verhindert werden, dass diese Resourcen auch anderweitig genutzt werden könnten. Desshalb appelliert diese Idee über den Umgang mit parasitären Methoden einmal mehr an unsere ethischen Prinzipien, welchen wir Folge leisten sollten.

4.4 Überwachung der Rechner im Netzwerk

Eine Variante, fehlbare Netzwerkteilnehmer zu identifizieren, könnte wie folgt beschrieben realisiert werden:

Bekanntlicherweise kommunizieren Rechner via einen oder mehrere Knotenpunkte innerhalb eines Netzwerkes miteinander. Jeder Knoten könnte seinen Verkehr kontrollieren und abnormale Aktivitäten wie beispielsweise auffallend viele ICMP Pakete eines einzigen Rechners oder einer Grupper von Rechnern einem speziell dafür eingerichteten Server melden, welcher die potentiellen „Hackermaschinen" entlarvt und entsprechende Alarme auslösen kann.

Fazit: Die Idee dieses Ansatzes verhindert zwar die Missnutzung fremder Rechenresourcen nicht, will aber einen Weg aufzeigen, wie Missbräuche von gutmütigen Mechanismen rasch aufgedeckt und entsprechend behandelt werden können.

4.5 Abschalten von nicht essentiellen Services

Eine wirksame Möglichkeit, sich gegen eine vielzahl von potentiellen Schwachstellen zu schützen, bietet die Variante, Services wie beispielsweise ICMP, welche nicht essentiell sind für die korrekte Übertragung von bestimmten Daten, gar nicht erst zur Verfügung zu stellen. Dies birgt zwar einerseits offensichtlich gewisse Nachteile mit sich, was etwa die Kontrolle von Netzwerkverbindungen anbelangt, andererseits aber kann so verhindert werden, dass triviale bösartige Programme einem Server ohne weiteres Schaden hinzufügen oder ihn lahmlegen können (siehe Abschnitt 3.3 Seite 7).

Fazit: Dieser Lösungsansatz bietet in seiner Idee keinen absoluten Schutz gegen sämtliche Missbräuche von Services, will aber aufzeigen, dass mit verkraftbaren Einschränkungen, welche im jedem Falle einen Komfort- und Funktionsverlust zur Folge haben, einiges unternommen werden kann gegen unerwünschten Resourcendiebstahl und weitere ungewollte, durch fremde Maschinen gesteuerte Aktionen.

5 Anhang

5.1 Abkürzungsverzeichnis

LAN	Abkürzung für „Local Area Network". Ein lokal angelegtes Netzwerk - im Gegensatz zu WAN, das überregional Arbeitsstationen und Netzwerke verbindet. „Lokal" bezieht sich in diesem Sinne auf einen gemeinsamen Standort, wie beispielsweise ein Firmengelände oder einen Raum.
MAC Adresse	Abkürzung für „Media Access Control". MAC wird im Netzwerk-Umfeld allgemein als „MAC-Adresse" einer Netzwerkkarte verstanden. Sie ist fest auf der Karte gespeichert und weltweit eindeutig; es handelt sich sozusagen um die unverwechselbare Seriennummer einer Netzwerkarte.
TCP/IP	Abkürzung für „Transmission Control Protocol/Internet Protocol". Bezeichnet zumeist die ganze Familie von Protokollen, die ursprünglich für das US-Verteidigungsministerium (Department of Defence - DoD) entwickelt wurden, um Computer in verschiedenen Netzwerken miteinander zu verbinden.
UDP	Abkürzung für „User Datagram Protocol". Bezeichnet ein übertragungsprotokoll. Es kann anstatt des TCP aus den TCP/IP-Protokollen verwendet werden, mit dem Unterschied, dass es nicht wartet, bis es eine Bestätigung erhält, ob ein Paket angekommen ist, oder nicht.
ICMP	Abkürzung für „Internet Control Message Protocol". Ist ein Nachrichten- und Fehler-Protokoll (Bestandteil von TCP/IP) zwischen Gateway und Host. Es ist jedoch nicht für den Benutzer erkennbar, sondern verrichtet seine Arbeit, nämlich eine fehlerfreie Übertragung zu gewährleisten bzw. den Sender über Probleme im Netzwerk zu informieren.
Internet	Das Internet ist eine dezentrales, weltumspannendes Netzwerk, d.h. es ist von keinem einzelnen Computer abhängig. Ursprünglich als ARPANet für das Militär in den USA entwickelt, ist es heute für Millionen Benutzer zugänglich. Das Netz besteht aus einer Reihe von Unternetzen, den Subnets; als Netzwerkprotokoll wird immer TCP/IP (Transmission Control Protocol / Internet Protocol) verwendet.

Erläuterungen entnommen von `http://www.computerlexikon.com`.

Literatur

[BFJB01] Albert-László Barabási, Vincent W. Freeh, Hawoong Jeong, and Jay B. Brockman. Parasitic computing. *Nature*, 412:894–897, August 2001.

[(CE02] Computer Emergency Response Team (CERT). Advisories and incident notes. `http://www.cert.org`, August 2002.

[Enz01] Christian Enzensberger. *Wir Parasiten. Ein Sachbuch.* Eichborn, Frankfurt am Main, 2001.

[Fer00] Paul Ferguson. Denial of service attack resources. `http://www.denialinfo.com/`, Juli 2000.

[For02] The Internet Engineering Task Force. Ietf website. `http://www.ietf.org/`, Juni 2002.

[Fre02] Vincent W. Freeh. Anatomy of a parasitic computer. *Dr. Dobb's Journal*, 332:63–67, Januar 2002.

[Lut01] Joerg Luther. Microsoft offline - anatomie eines gaus. `http://www.tecchannel.de/internet/640/`, Januar 2001.

[Oeb00] Alfons Oebbeke. E-mail-virus: I love you. `http://sicheres.web.glossar.de/glossar/1frame.htm?http%3A//sicheres.web%.glossar.de/glossar/z_loveletter.htm`, Mai 2000.

[SET01] SETI@home. The search for extraterrestrial intelligence. `http://setiathome.ssl.berkeley.edu/`, 2001.

[Ste94] Richard W. Stevens. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, 1994.

[Wir95] Niklaus Wirth. *Digital Circuit Design*. Springer Verlag, August 1995.