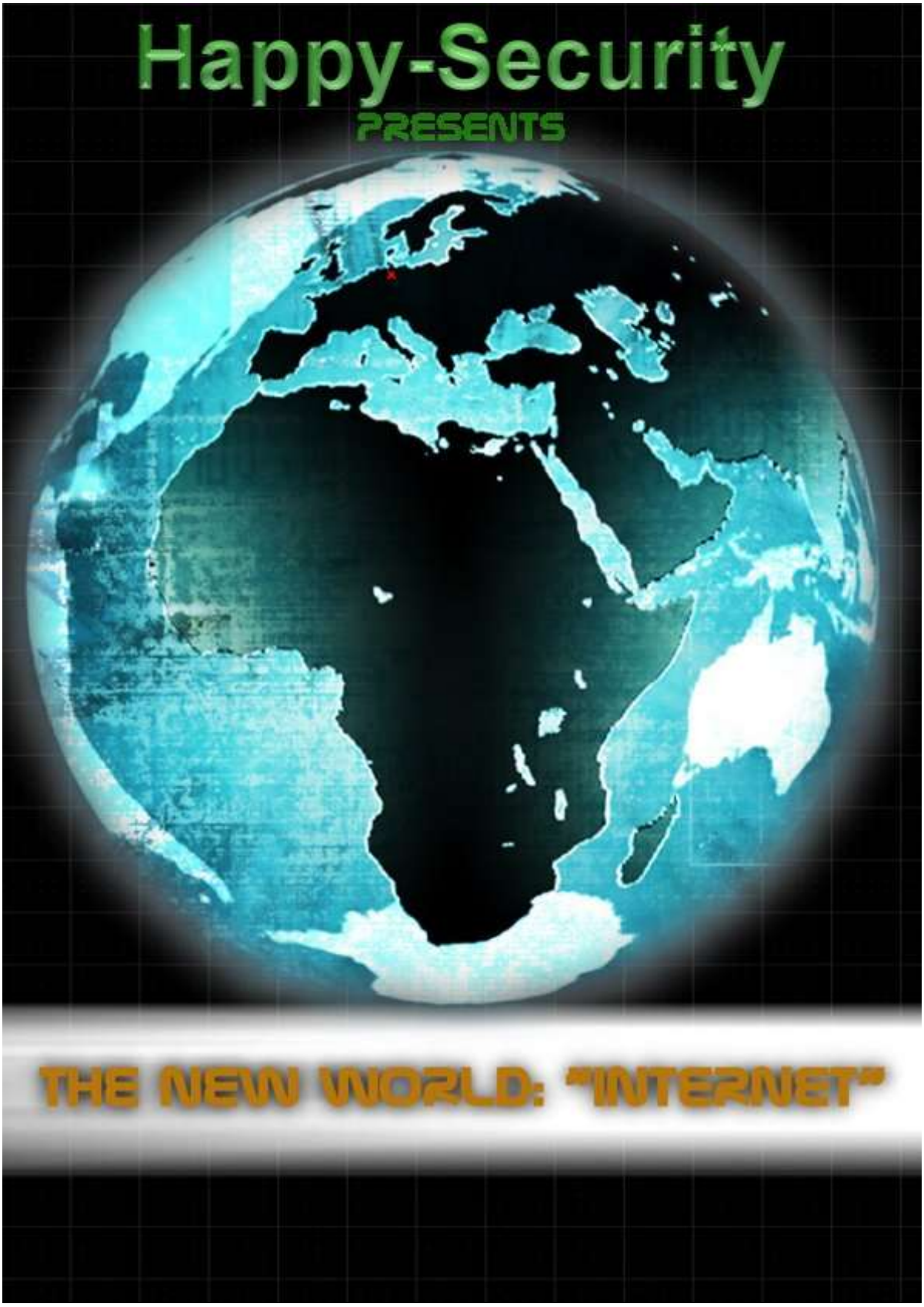


The New World: INTERNET

by

Tsutomu Katsura

[www.happy-security.de]

Inhaltsverzeichnis:

Einleitung, die Erste

Seid gegrüßt ihr Cyberkrieger und alle die es werden wollen. Zu aller erstes werde ich einmal erläutern, aus welchem Grund ich überhaupt solch einen Text hier verfasst habe. Einige unter euch werden sich wahrscheinlich wieder denken: "Was soll das ganze? So etwas gibt es doch schon Haufenweise im Netz"... und zu dieser These kann ich nur mit dem Kopf nicken! Es stimmt... doch frage ich mich auch oft wenn ich mal wieder im Netz unterwegs bin: "Wie kann es sein, dass obwohl mit diesen Texten, Büchern, News-Sites (Heise.de) oder ähnlichem die Internet-Nutzer immer noch so unvorsichtig sind? Oder gar keine richtige Vorstellung von den Dingen haben, mit denen sie arbeiten?“

Es ist schon traurig mit anzusehen, wie einfach irgendwelche Script-Kiddies mit herunter geladenen, kleinen Programmen unbefugt Zugriff auf Firmen-/Schulserver kommen und andere Leute nerven. Doch liegt die Ursache nicht nur in der Beschaffung solcher Programme, sondern viel mehr liegt die Ursache für leichte Angreifbarkeit bei den Benutzern und Programmierern der Soft-/Hardware. Viele Privatanwender installieren sich ein Betriebssystem (meist Windows 98/XP) und gehen dann über ihre Telephonleitung ins Netz. Betrachtet man diesen Rechner 2-3 Monate später noch einmal, so muss man leider verstellen, dass sich nix getan hat. Ok, bis auf ein paar Dialer, Trojaner und Spyware Zeugs hat sich nichts geändert. Die gesamte Zeit wurde kein Sicherheitsupdate durchgeführt (was bei Microsoft™ meiner Meinung nach ungefähr alle 5 Tage geschehen sollte, bei der Menge an Bugs, etc), immer noch keine Firewall installiert und wohl wurde bei der Konfiguration der Konten kein wirklich sicheres Passwort gewählt. WARUM das so aussieht? Weil in den Medien wohl immer noch nicht genug gewarnt wird oder besser weil dem Benutzer diese Warnungen falsch übermittelt werden. Es ist schön, dass immer gewarnt wird, aber wofür genau? Im Grunde wird immer nur empfohlen, man solle doch aus dem Internet so oft es geht ein Virus-Update durchzuführen... Wichtige andere Vorkehrungen um es gar nicht erst so weit kommen zu lassen, werde fast nie erwähnt.

Genau darum geht es in diesem Text! Es soll noch einmal mehr auf die aktuellen Sicherheitslücken aufmerksam machen, so dass vielleicht bald alle verstehen, wieso Internet-Sicherheit in unserer Zeit viel mehr Beachtung geschenkt werden muss. Es wird hier jedoch versucht so gut es geht auf kompleze Prazisbespiele zu verzichten, da dies den Rahmen dieses „Buches“ sprengen würde. Sollten jedoch trotzdem Praxisbespiele erwünscht sein, so schaut doch einfach auf unserer Website www.Happy-Security.de vorbei und fragt in der Community. Ich bin sicher, dass euch dort mit Rat und Tat gehölfen wird.

Zweite Einleitung (Deja vu)

Auch wenn es nun das Internet schon einige Jahre gibt, so muss man sich doch immer noch wundern, wie viele Leute ohne Absicherung ihrer Systeme im Internet surfen. Sogar große Firmen sparen an spezialisiertem Personal, bis es dann zu spät ist und der komplette Server für mehrere Stunden offline ist. Die dadurch entstandenen Kosten (bei einem Online-Shop zum Beispiel) liegen oft bei mehreren tausenden Euro pro Stunde! Meistens werden solche Websites von geschulten Administratoren überwacht und verwaltet, die hoffentlich genügend Erfahrungen mit dieser Materie haben.

Dieses Buch soll aus Gründen der Komplexität auch nicht an geschulte Administratoren oder ähnliches gerichtet sein, sondern viel mehr unseren Besuchern der Happy-Security.de Website als Basis-Kompendium dienen. So ist dieses Werk auch nur als leichter Lesestoff zu verstehen, mit welchem es im Nachhinein einfacher fallen soll, sich später auf andere (nicht so einfache) Texte zu stürzen, ohne schon beim Lesen des ersten Satzes zu sagen, „ich verstehe das nicht“!

Wenn also jemand aus dem Chat oder aus unserem Forum auf dieses Werk verwiesen wurde, dann könnt ihr euch sicher spätestens an diesem Zeitpunkt denken, was der Grund dafür war. Ihr sollt euch selbst mit der Thematik befassen und alles, was ihr zu Anfang braucht, findet ihr in diesem Text und google.

Was genau sind eigentlich Hacker?

Als ich vor einigen Jahren einmal eine Umfrage in unserer Schule gestartet habe was denn ein Hacker sei, musste ich oft die Antwort hören: "Es seien Leute, die Viren schreiben und damit Computer kaputt machen." und: "Die sind im Internet und hacken sich in Firmenwebsites rein, um dort Daten zu stehlen." Diese Meinung vertraten ausgerechnet rund 90 % der Befragten. Was für ein schockierendes Ergebnis. Dies war ein Grund mehr der Öffentlichkeit zu zeigen, was sich wirklich dahinter versteckt.

Auch wenn im Gegensatz zu früher der Begriff Hacker in den Medien und in den Filmen sich stark gewandelt hat, so ist das Grundprinzip noch immer erhalten worden. Heutzutage ist in den Medien ein Hacker ein Mensch, der sich Zugriff zu einem System verschafft, um dort Unruhe zu stiften. Im Grunde stimmt dies ja auch noch, doch ist dies nur ein kleiner Teil...

Meiner Meinung nach ist ein Hacker ein Computer-Freak, welcher die Welt mit anderen Augen sieht als der Rest der Menschen. Er arbeitet viel mit seinem Computer und versucht sein Gelerntes noch zu steigern, um besser zu sein als die anderen. Hacken fängt da an, wo man mehr tut als im Benutzerhandbuch steht. Das Eindringen in andere Rechner hat dabei eigentlich nur zwei Gründe:
1. Ausprobieren wie weit man mit seinem Wissen kommt (was man mit seiner Maschine alles anstellen kann...)
2. Informationsfreiheit. Ein Punkt der Hacker-Ethik ist es: Öffentliche Daten für alle zugänglich zu machen. Damit nicht nur ein Teil der Bevölkerung daraus Vorteile ziehen kann.

Ursprünglich war der Begriff Hacker aber nicht so negativ bewertet. Als Hacker bezeichnete man einfach jemanden, der unermüdlich arbeitete, um zu einem Ergebnis zu kommen, wie dies bei den ersten Hackern des MIT (Massachusetts Institute of Technology) der Fall war. Diesen Hackern haben wir es zum Beispiel zu verdanken, dass es auch noch eine Ausweichmöglichkeit zu Windows gibt. Die Rede ist von UNIX! Außerdem wäre ohne diese Leute das Internet auch nicht das was es heute wäre. Wenn man den Begriff heute noch in diesem Zusammenhang gebrauchen würde, so müsste man sogar sagen, dass Linus Torvalds auch ein Hacker ist, denn er hat den ersten lauffähigen Linux-Kernel entwickelt. Im Gegensatz dazu stehen heute die Cracker... Sie versuchen sich um jeden Preis Zugriff zu einem System zu verschaffen und es ist ihnen oft egal, ob sie dann bei der Spuren Verwischung sensible Daten löschen oder gar das gesamte System zerstören (meist ist dies sogar ihr eigentliches Vorhaben). Natürlich gibt es auch noch die alten Cracker-Groups (und deren Nachfolger), welche sich darauf spezialisiert haben Kopierschutz von Software zu umgehen und diese dann im Internet zu verteilen. Ihr Ziel ist es, ähnlich dem

Hacker, Anerkennung zu bekommen für ihre geleistete Arbeit. Wenn ein neues Software-Produkt auf dem Markt erscheint, so wird versucht diese noch am gleichen Tag geknackt online zu stellen (o-day Warez). Auch haben sich früher solche Gruppe zusammen getan um coole Grafiken zu gestalten (als Intro für ihre gecrackte Software oder einfach so). Als dann der Kopierschutz zunehmend komplexer wurde und die Gesetze verschärft wurden, haben sich viele Cracker abgeseilt und sich nur noch auf das erstellen solcher Grafiken spezialisiert (wie zum Beispiel die Visualisation in diversen Multimedia-Playern). Ich werde jetzt aber nicht weiter auf diese Gruppe von Computer-Freaks eingehen, sondern mit den Hackern weitermachen.

Was hat es mit der Hacker-Ethik auf sich?

Vor vielen Jahren, als die Computergesetze verschärft worden und es einen kleinen Krieg in der Hackerszene gab, wurde das Manifest verfasst. Geschrieben wurde es von "The_Mentor" (Mitglied der Gruppe "LoD") kurz bevor er von der Polizei festgenommen wurde. Mittlerweile ist es um den gesamten Globus gewandert und in zig verschiedene Sprache übersetzt worden. Natürlich gibt es auch einige in der Szene, denen dieses Manifest, sowie die Hacker-Ethik am Arsch vorbei geht, aber dennoch versuchen viele sich nach den Worten der Verfasser dieser Texte zu richten. Es muss jedoch wichtiger Weise einmal darauf hingewiesen werden, dass es sich nicht um Vorschriften oder Gesetze in dieser Schriften handelt, sondern dass es viel mehr ein Leitfaden sein soll.

"Dies hier ist nun unsere Welt... die Welt des Elektrons und der Schaltungen, der Schönheit des Baud. Wir benutzen eine Dienstleistung, die eigentlich Spottbillig wäre, wenn nicht ein Haufen profitgieriger Säcke sie anbieten würde... und ihr nennt uns Kriminelle.
Wir erforschen... und ihr nennt uns Kriminelle.
Wir streben nach Wissen... und ihr nennt uns Kriminelle.
Wir existieren ohne Hautfarbe, Nationalität oder religiösen Vorurteilen.

Ihr baut Atombomben, führt Kriege, mordet, lügt und versucht uns klar zu machen, dass es das Beste für uns wäre und dass ja die Kriminellen seien.

Ja, ich bin ein Krimineller. Mein Verbrechen heißt Neugier. Mein Verbechen ist es, Leute danach zu beurteilen, was sie sagen und denken, und nicht danach, wie sie aussehen. Mein Verbrechen ist es, dass ich cleverer bin als ihr, etwas, was ihr mir nie verzeihen werdet.

Ich bin ein Hacker, und dies ist mein Manifest. Mag sein, dass ihr mich aufhaltet, aber ihr könnt uns nicht alle aufhalten... Denn am

Ende sind wir alle gleich!"

Da es nun schon lange Hacker in Medien gab, haben sich einige von Ihnen zu Gruppe zusammen geschlossen um die Ziele der Hacker zu veröffentlichen und weltweit zu vereinheitlichen. Dies wurde dann zu der bekannten HACKER-ETHIK:

- Der Zugang zu Computern und allem, was einem zeigen kann, wie diese Welt funktioniert, sollte unbegrenzt und vollständig sein.
- Alle Informationen müssen frei sein.
- misstraue Autoritäten - fördere Dezentralisierung.
- Beurteile einen Menschen nach dem, was er tut und nicht nach üblichen Kriterien wie Aussehen, Alter, Rasse, Geschlecht oder gesellschaftlicher Stellung.
- Man kann mit einem Computer Kunst und Schönheit schaffen.
- Computer können dein Leben zum Besseren verändern.
- "Mülle" nicht in den Daten anderer Leute.
- Öffentliche Daten nützen, private Daten schützen.

1337-H4X3Rz

Eine lustige Einführung von einigen Hackern war die 1337-Schrift. (1337=Leet) und das Verwenden von Synonymen in ganz normalen Sätzen. Hacker (oder besser: extreme Computer-Freaks) denken anders und sehen diese Welt mit anderen Augen. Wenn man den ganzen lieben Tag mit Computern arbeitet, entwickelt man über kurz oder lang eine von anderen Menschen teilweise sehr stark abweichende Lebens- und Denkweise. Dies ist bei mir in keinster Weise anders *smile*.
Zitat: "Bei solchen Smaltalks vertreibt man sich auch gerne die Zeit mit lustigen Bauernweisheiten. Wer aber sin Haus ohnehin nur kurz verlässt, hat wenig Interesse über das Wetter zu meckern als z. B. über sein Betriebssystem, was mal wieder nicht so will wie man selbst. ... Mag sich der eine über Abkürzungserklärungen wie *Einschalten, Ausschalten, Geht nicht!* oder *Auf Eigene Gefahr!* für AEG erfeuen, so hat der andere eben mehr Spaß an *Dosen*, *Winblows* oder *AOHell*."

Ebenso sieht man seine Umgebung etwas anders als hätte man es früher getan. So wundert man sich plötzlich über die geile Auflösung dieser sogenannten Realität wenn man aus dem Fenster schaut. Auch überlegt man sich wenn man einmal wieder durch Gegend schländert und einen Gegenstand sieht, der einem gefällt, wie man diesen grafisch mit PhotoImpact erstellen und noch bessere Lichteffekte einbauen könnte.

Viele Leute, denen man begegnet, erzählen einem welchen Star sie gerade mögen... Fragt man dann einmal etwas genauer nach, so erhält man meist nur eine einzige Antwort: "Der ist ja so süss!" Natürlich ist mir dieser Punkt nicht vollkommen egal, aber es kommt doch wohl viel mehr darauf an wie sich eine Person verhält und was diese Person

für eine Einstellung zu verschiedenen Themen vertritt. Als Beispiel wäre dort der Musiker und Rapper Eminem zu nennen. Er ist einer der sich von den Rest der Musikindustrie abgewendet hat, weil er etwas sagen will mit seinen Texten und nicht nur kommerz produziert um Geld zu bekommen. Dabei spielt es jetzt einmal keine Rolle, wie er auftritt... Fakt ist, dass er seine Meinung publik macht, frei sagt was er denkt und dabei nicht um den heißen Brei redet. Daher kommt es auch heute viel mehr darauf an, was Menschen im Internet schreiben und wie sie es wiedergeben. Um sich ein wenig von den anderen Surfern abzugrenzen und um seinen Gebenüber zu zeigen dass man smarter ist, haben Hacker sich überlegt eine Schreibweise zu wählen, die nicht jeder versteht, um schon einmal im Vornherein sicher zu stellen, mit wem man es zu tun hat.
Es werden dabei einfach einzelne Buchstaben mit anderen Zeichen ersetzt, um das Aussehen zu verändern aber dennoch den Sinn des Sätzes zu bewahren. So kann man zum Beispiel für "Computer" auch "<0|V|pu74r" schreiben.
Auch haben Hacker oft einen überdurchschnittlichen Wortschatz, da sie um sich um ihr Wissen anzueignen viel lesen (was heutzutage ja nicht mehr so der Fall für viele unter 30 ist) und außerdem oft zu Rollenspielen trifft. (nicht diese Computergames, sondern man trifft sich, setzt sich an einen Tisch und versetzt sich in die Gedanken seines Charakters, mit welchem er Abenteuer zu bestehen hat).

Entstehung des Internets

Am Anfang war das ARPANET... In diesem weltweitem Netzwerk waren zu anfang nur einige sehr leistungsstarke Rechner von Universitäten und Militär.
An Websites geschweige denn Tauschbörsen wie zum beispiel eDonkey war da noch überhaupt nicht zu denken. Zur Verfügung standen nur Telnet und FTP, welche auf Basis des Netzwerkprotokolls NCP entwickelt wurde. Sechs Jahre später wurde anstelle von NCP das TCP-Protokoll eingeführt. welches auch heute noch benutzt wird. Als dann einige Jahre darauf, besser gesagt 1984, auch noch das Usenet dazu kam, wurde das ARPANET auch für privat Personen nutzbar. Nun waren ungefähr ein tausend Rechner vernetzt. Da nun die Identifikation der einzelnen Rechner immer schwerer wurde, wurde ein neues System dazu entwickelt. Ich spreche hierbei von DNS (Domain Name Service), wobei jeder Rechner eine eindeutige IP-Adresse bekam. Diese Technik wird heute auch noch verwendet.

- 1988 wurde von einige Programmierern IRC entwickelt.
- CompuServe hat sein Mail-Service ins Internet gestellt.
- 1990 waren rund 100.000 Rechner im Internet vertreten.
- Archiv-Server wurden als Suchmaschinen eingesetzt.
- In Mail konnten Attachenments angehängt werden (MIME).
- 1993 Entstehung von HTTP!

Nun konnten die einzelnen Server sich bei INTERNIC anmelden um eine IP-Adresse zu bekommen. Mit dieser Adresse konnte man nun eine Website zur Verfügung stellen, die auf Basis von HTML programmiert wurde. /* Anmerkung: HTML ist eigentlich keine richtige Programmiersprache... eher eine Formatierungssprache '/ Um diese Website anschauen zu können, benötigt man zu allererst einen Browser. Der erste entwickelte Browser, der HTML lesen konnte war Mosaic. Ein Jahr später folgen schnell mehrere Alternativen zu Mosaic. Einer von ihnen war der Netscape Communicator. Microsoft's Internet Explorer zog erst ein Jahr später auf die Festplatten der User, da Microsoft keine Dringlichkeit darin sah Software fürs Internet herzustellen. Tja, selbst schuld Billy! Erst 1997 (mit Einführung von Office 97) hat Microsoft sich richtig ums Internet gekümmert.

Dienste der Internets

In diesem Kapitel möchte ich einmal kurz auf die wichtigsten Dienste des Internets weiter eingehen. Diese wären: HTTP, FTP, SMTP, POP3, IMAP, IRC, Finger & Telnet...

HTTP müsste eigentlich jeder von euch kennen oder wenigstens schon mal benutzt haben. HTTP steht für Hyper Text Transfer Protokoll. Eng damit verbunden ist das HTML. Dies ist für das Betrachten von webseiten äußerst wichtig. Übersetzt bedeutet es so viel wie: "Text-Formatierungssprache". Dieser Dienst wird benötigt um im Internet auf Homepages zuzugreifen. Dazu verwendet man eine Web-Browser... Das sollte ja eigentlich bekannt sein, sonst solltest du vielleicht einmal ein Buch von "Computer-Bild" lesen *grins*. Früher konnte nur ganz einfacher Text interpretiert werden. Dann wurde das Protokoll erweitert, so dass auch Bilder und andere Text-Formatierungen vorgenommen wurden konnten. Später dann kamen noch andere Erweiterungen hinzu. Diese sind zum Beispiel JavaScript, CGi, ASP und PHP. Durch diese Script-Sprachen wurde es möglich gemacht, dass nun programmierspeziefische Funktionen eingesetzt werden können (Variablen, if-then-else, schleifen, etc). Durch die Erweiterung der Angaben, Header-Information und Fehlercodes, ist es allerdings mehr als nur auf Hyper-Text beschränkt. Wenn man jetzt also eine Datei auf einer Webseite über die URL aufruft, so landet beim Server die Anfrage nach dem Dokument. Die Anfrage sieht ungefähr so aus:

```
GET /index.html HTTP/1.1
Host: www.happy-security.de
```

Hier bekommt unser Server die Anfrage, doch bitte mal die Datei index.html an den Absender der Anfrage zu schicken. Die Antwort kann dann so aussehen (positive Rückmeldung):

```
HTTP/1.1 200 OK
Server: Apache/2.0 (Unix) PHP/5
Content-Length: 3200
Content-Language: de
Content-Type: text/html

(Inhalt von infotext.html)
```

Nun erhalten wir als erstes ein paar Informationen zum Dokument und zum Server. Das Erste (*200 Ok*) gibt an, dass die Operation erfolgreich war und alles ok ist. Wie sicher bekannt, gibt es ja dann noch zum Beispiel die sogenannten 404-Fehler. Diese treten auf, wenn eine Datei nicht gefunden worden ist auf dem Server. HTTP-Statuscodes kann man auf http://netzikon.net/misc/http-statuscodes.html nachlesen.
Als nächstes bekommen wir die Info, um was für einen Server es sich

handelt und welche PHP Version dort benutzt wird. Für das normale Surfen ist dich jedoch gänzlich uninteressant ^_^ Für Penetrationstests jedoch umso wichtiger. *Content-Length* beschreibt die Byte-Größe der angeforderten index.html . Darauf folgt dann noch die Sprache und Art der Datei. Hier ein einfaches HTML-Dokument. Nach nun einer Leerzeile geht es los mit der eigentlichen Datei, die wir angefordert hatten. Diese wird dann von unserem Browser eingelesen, verarbeitet und ausgegeben.

Da jedoch eine normale HTTP-Anfrage unverschlüsselt zum Server gesendet wird, ist das Besuchen des Bankkontos von zu Hause keine gute Idee... jedoch im Netzwerk könnte einfach die gesendeten GET-Requests mitlesen. Hierfür wurde jedoch schon seit längerer Zeit auch eine Lösung. HTTPS (Hypertext Transfer Protocol secure) baut nicht sofort über TCP eine Verbindung zum Server auf, sondern verschlüsselt vorher die Daten über SSL/TLS. Das Prinzip funktioniert ähnlich wie PGP. Es wird vom Server vorab ein Zertifikat angefordert und der öffentliche Schlüssel ausgetauscht. Hier möchte ich jedoch nicht weiter darauf eingehen, da dies zu detaillastig wäre.

Weitere Infos dazu Stichwort: Diffie-Hellman-Schlüsselaustausch

FTP hingegen ist ein nicht-grafischer Dienst... FTP steht für "File-Transfer-Protokol". Mit diesem Dienst ist es möglich sich die Baumstruktur des Servers, beziehungsweise des gehosteten Speicherplatzes seiner Domain, in einem Programm anzeigen zulassen. Früher wurde das ganze ganz einfach mit dem Befehl "FTP 192.168.0.1" in der Console ausgeführt. Wenn man den Anfangsbefehl eingegeben hat, so muss man sich bei dem Server erst einmal authorisieren. Dies geschieht mit Benutzernamen und dem dazu gültigen Password. Sobald man den Benutzernamen und das Passwort eingeben hat, so wird in "etc/passwd" kontrolliert, ob die Angaben übereinstimmen. Ist dies der Fall, so wird man am gültiger User eingeloggt und erhält die für seine Gruppe (group) aufgeführten Rechte. Da die meisten Server unter Unix laufen, ist die Einstufung der Gruppenrechte sehr wichtig. Man kann sich auf den meisten Server sogar als Gast einloggen, sofern man keinen Account besitzt. Dies hat jedoch den Nachteil, dass man so gut wie überhaupt keine Rechte hat. Man darf vielleicht ein paar öffentliche Dokumente (meist im Ordner "public") lesen, aber nicht auf Daten von anderen Benutzern zugreifen, nichts ausführen und schon gar nichts löschen. Trotzdem ist dieser Gast-Service ein gutes Hilfsmittel für Hacker. Wenn man nach dem Benutzernamen gefragt wird, so muss man einfach "anonymous" eintippen. Wenn dann die Passwortabfrage kommt, so soll man eigentlich seine E-Mail Adresse angeben... die meisten geben jedoch eine gefakete an, da es sowieso nicht überprüft wird (Sehr beliebt ist: bgates@microsoft.com *grins*).

Bei **SMTP** handelt es sich um ein Protokoll für die Übertragung von E-Mail Nachrichten. Dieses Protokoll wird in **verschiedenen** Mail-

Programmen benutzt (so zum Beispiel in Outlook Express). Durch Fehler in Einstellungen vom Server, kann ein anonymer Benutzer unter Umständen Mails mit gefälschtem Absender versenden. Bei **POP3** werden die Nachrichten nur vom E-Mail Server abgeholt und local bzw. bei einem WebMail-Account gespeichert. Die Serverseitige Variante heißt **IMAP**. Bei **IMAP** werden also die Nachrichten auf dem Mail-Server gelesen, versendet und bearbeitet.

Als letzten wichtigen Dienst haben wir dann noch **TELNET** (Terminal emulation Net). Mit Telnet lässt sich einiges Anfangen. Am Anfang war Telnet nur dazu gedacht um entfernte Systeme zu verwalten. Wenn man zum Beispiel als Administrator in Hamburg tätig ist und gerade Urlaub in Japan macht, so kann man sich mit **TELNET** ganz einfach bei dem Server in Hamburg anmelden, um daran zu arbeiten. Hierfür werden die SMTP Befehle verwendet. Der Befehl um sich bei dem Rechner einzuklinken lautet: "telnet <IP-Adresse> <Port>" Wobei die Angabe für normale Arbeiten ruhig weggelassen werden kann. Standardmäßig ist Telnet Port 23. Man hat jedoch die Möglichkeit sich mit anderen Diensten verbinden zu lassen. Sobald man dann den Befehl eingetippt hat, kommt erst einmal eine Abfrage nach der Benutzerkennung und dem dazugehörigen Passwort. Wichtig ist hierbei natürlich das der User im TELNET-Server existiert und für die TELNET Benutzung freigeschaltet wurde.

In all diesen Fällen gibt es Schwachstellen, die dazu ausgenutzt werden können, um ein System lahm zu legen. Diese Schwachstellen werden in der Szene meist als Exploit bezeichnet (auch Xploit genannt). Später werde ich noch weiter darauf eingehen und genauer zeigen was mit diesen Schwachstellen angestellt werden kann.

TCP/IP

Dort wo der ganze Verkehr lang fließt, befindet sich das TCP/IP. Die Abkürzung „TCP“ steht für Transmission Control Protokoll. Die Abkürzung für „IP“ steht für Internet Protokoll. Diese beiden Protokolle wurden in den 70er Jahren vom amerikanischem Verteidigungsministerium entwickelt und fand in UNIX-Systemen starke Verbreitung und ist mittlerweile das weitverbreitetste Übertragungsmodul der Vermittlungsschicht (Transport). Da wie viele andere Protokolle das TCP/IP-Protokoll von der ISO genormt worden ist, sind alle Protokolle miteinander kompatibel. Das heißt auch, dass es Plattformunabhängig ist und somit auf Windows, MAC, sowie auf Unix-Systemen gleichermaßen genutzt werden kann. Nur die Kommunikationsprogramme, die die Daten umwandeln und wiedergeben, sind vom Aufbau etwas unterschiedlich. Dies macht jedoch keinen Unterschied für die einzelnen Protokolle. Die meisten Dienste im Internet bedienen sich des TCP/IP-Protokoll zur sicheren Datenübertragung. So zum Beispiel auch die bereits oben erklärten Dienste. So viele Dienste über dieses Protokoll laufen zu lassen ist nur möglich, da im Grunde TCP/IP aus einer ganzen Familie von Protokollen besteht, die modular miteinander verknüpft sind und in verschiedenen Schichten aufeinander aufbauen. Insgesamt sind es vier von den sechs Schichten, in denen das TCP/IP-Verfahren Verwendung findet. Die Zusammensätzung der Architektur sieht grundsätzlich so aus:

-> Netzwerkschicht
-> Internetschicht
-> Transportschicht
-> Anwendungsschicht

In der Anwendungsschicht befindet sich neben dem TCP auch noch das UDP (User Data Protokoll). Zusammen ist es möglich über diese Protokolle und den bekannten Ports (FTP, HTTP, SMTP) mehrere Anwendungsprogramme zur selben Zeit laufen zu lassen. Außerdem befindet sich hier der Domain Name Service (DNS), welcher den einzelnen Schnittstellen im Netzwerk/Internet einen Namen vergibt. Ohne diesen Dienst, müsste jeder IP-Adressen eintippen um eine Website zu besuchen. Anstelle www.linux.com müsste man zum Beispiel 111.25.172.3 eintippen. Das wäre ziemlich schlecht zu merken.

In der Netzwerkschicht müssen auf die Einhaltung der Details der anderen Schichten geachtet werden. Das heißt, es muss auf die richtige Adressierung in den jeweiligen Netzen und auf die Paketgröße achten. Hinzu kommt, dass bei neuer Technologie der Übertragung im Hardwarebereich, neue Protokolle installiert werden müssen. Die Netzwerkschicht wird auch oft Sicherungsschicht des OSI-Modells genannt.

Die Internetschicht hingegen wird für das richtige Zustellen von IP-Paketen auf Internetbasis eingesetzt. Bei der Übertragung gehen die Pakete ja per Routing über viele verschiedene Teilnetze. Ohne die Internetschicht und dem darin Implementierten ICMP-Protokoll, wäre es nicht möglich eine Antwort von einem entfernten Rechner zu erhalten. Das ICMP-Protokoll dient zur Übertragung von Diagnose- und Fehlerinformationen in dieser Schicht.

Das wichtigste in der Transportschicht ist, dass beim Übertragen von Datenpaketen immer eine Sequenznummer mit gesendet wird. Diese Sequenznummer soll das doppelte Empfangen der Pakete unterbinden und den Empfang von richtig zu gestellten Daten quittieren.

Aufbau von Paketen

Ein Paket setzt sich aus Header (20-Byte-Header) und Datenblöcke zusammen. Genau genommen handelt es sich nicht um „Paket“ sondern um ein „Segment“ im TCP-Protokoll.

<table>
<tr><td colspan="2">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td colspan="4">Sequenz number</td></tr>
<tr><td colspan="4">Acknowledgement number</td></tr>
<tr><td>Data Offset</td><td>Reserved</td><td>URG, ACK, PSH,
RST, SYN, FIN,</td><td>Window</td></tr>
<tr><td colspan="2">Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="4">Options</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

Solch ein Paket wird vom Client zum Host über das Netzwerk/Internet versendet. Angenommen man geht auf winzip.com und holt sich dort das neueste Update... damit der Download schneller und sicherer funktioniert, werden die Daten in solche Pakete von 65.545 Bytes ausgeteilt. Diese ganzen Informationen im Header sind wichtig, um die geteilten Pakete am Ende (also wenn der Download fertig ist) wieder zusammenzufügen.

Erläuterung der einzelnen Begriffe:

Source Port
(16 Bit) Hier ist der Quellport der Verbindung gespeichert.

Destination Port
(16 Bit) Zielport der Verbindung

Sequenz number
(32 Bit) Jeder Datenblock bekommt beim Versenden eine Sequenznummer. Wenn dann alles angekommen ist, muss der Rechner diese Blöcke wieder rückwärts zusammenfügen. Diese Zahl darf sich natürlich nicht wiederholen, da sonst falsch zusammen gewürfelt würde. Aus diesem Grund wird die Sequenznummer immer ein inkrementiert.

Acknowledgement number
(32 Bit) Dies ist das Gegenstück für den Empfänger. Es gilt als Bestätigung um zu kontrollieren, welche Pakete mit welcher Nummer schon gesendet worden sind. Kommt die Bestätigung, dass Sequenz 1234 empfangen wurde, so gelten die älteren Blöcke auch als bestätigt.

Flag Code
Die kleinen Felder URG, ACK, PSH, RST, SYN und FIN sind Zustandsfelder. Sie können entweder 1 oder 0 als Wert besitzen und sind deshalb jeweils 1 Bit groß.

Flag	***Name***	***1***	***0***
URG	Urgent	Schnelle Versendung	Normale Sendung
ACK	Acknowledge	ACK-Number ist gültig	ACK-Num ignorieren
PSH	Push	Bei Ankunft werden die Daten bereitgestellt.	Daten werden gepuffert
RST	Reset	Bei Fehler wird Verbindung zurück gesetzt.	-
SYN	Synchronize	Drei-Wege-Handshake	-
FIN	Finish	Beendet die Verbindung	-

Checksum
Die Checksum-Funktion prüft alle vorangegangenen Informationen durch Prüfsummenberechnung. So wird sichergestellt, das alles Daten korrekt sind und kein Paket falsch empfangen/gesendet wurde.

URGent Pointer
Ist in diesem Feld ein Wert gesetzt, so werden bestimmte, als dringend markierte, Daten sofort gesendet und ausgewertet.

Durch diesen festgelegten Aufbau von TCP-Paket wird gewährleistet, dass die Daten korrekt und in richtiger Reihenfolge den Zielrechner erreichen. Beim Empfäng sowie beim Senden, wird in mehrfachen Operationen überprüft, ob das Paket schon einmal existiert (Sequenznummer), alles unverfälscht übertragen wurde (Prüfsumme) und ob Daten schnell gesendet werden müssen. Ein Paket wird so lange versucht zu senden, bis die Gegenstelle den korrekten Empfang mit einem ACK-Paket bestätigt oder bis ein Time-Out eintrifft.

Sockets (IP+Port)

IP

Um im Internet oder Netzwerk die Daten richtig zu versenden, muss jeder Rechner eine eigene Adresse besitzen. Diese wird als IP-Adresse bezeichnet. Sollte im Netzwerk ein zweiter Rechner mit der selben IP gefunden werden, so kommt es entweder zu Fehlern oder der letztere Rechner bekommt kein Zugang. Zwar kann man jeden Rechner mit einem einfacher zu merkenden Namen versehen, doch auf der Protokollebene werden diese Namen wieder in die dazugehörige IP umgewandelt. Im Netzwerk lassen sich die IPs selber bestimmen. Im Internet jedoch werden die Nummern von einer zentralen Stelle zur Verfügung gestellt (IANA). Internet-Provider kaufen der IANA eine bestimmte Anzahl an IPs ab, die Sie jedem ihrer User, der online geht, verteilen können.

Grundsätzlich lassen sich IP-Adressen jedoch auch noch mal in verschiedene Gruppen (besser gesagt: Klassen) aufteilen. Insgesamt sind es 4 Klassen, die für verschiedene Nutzungsarten benutzt werden können.

```
CLASS-A-Network:   1.nnn.nnn.nnn – 126.nnn.nnn.nnn
     =>        126 Netze für    16.777.216 einzelne Rechner
CLASS-B-Network: 128.nnn.nnn.nnn – 191.nnn.nnn.nnn
     =>     16.384 Netze für        65.536 einzelne Rechner
CLASS-C-Network: 192.nnn.nnn.nnn – 223.nnn.nnn.nnn
     => 16.777.215 Netze für 4.261.412.610 einzelne Rechner
CLASS-D-Network: 224.nnn.nnn.nnn – 255.255.255.0
```

insgesamt lassen sich also 4.294.967.296 Rechner an das Internet anschließen... Wohlgemerkt: Gleichzeitig, da nur IPs vergeben werden, wenn ein Rechner im Internet ist. IP-Adressen sind bis zur IP-V6 immer 32 Bit lang. Jede der vier Zahlen repräsentiert 8 Bit, also 1 Byte.

```
204.152.190.21 => Altavista.com
204.152.190.0  => Netz in welchem sich Altavista.com befindet
204.152.190.255=> Broadcast-Adresse dieses Netzes
```

Mit der Broadcast-Adresse lassen sich alle Rechner in einem Netz ansprechen. Deshalb ist diese oft bei Schwachstellenausnutzung zu berücksichtigen, dass alles korrekt konfiguriert worden ist.

Port

Sind die Daten nun auf dem Zielrechner angekommen, müssen die Daten jedoch noch ein bisschen weiter zur jeweiligen Anwendung (FTP, HTTP, telnet, etc.) geleitet werden. Dies funktioniert durch die einzelnen Ports der Anwendungen...

1	tcpmux	101	hostnames
3	cfinger	109	pop-2
7	echo	110	pop-3
9	discard	130	icp
11	systat/isdnlog	137	netbios-ns
12	vboxd	138	netbios-dgm
13	daytime	139	netbios-ssn
15	netstat	150	ninstall
19	chargen	194	irc
20	ftp-data	220	imap3
21	ftp	443	https
22	ssh	512	exec
23	telnet	513	login
24	private	514	shell
25	smtp	515	printer
36	ssl-ldap	526	tempo
37	time	529	support
42	nameserver	543	klogin
43	whois	544	kshell
53	domain	557	fax
70	gopher	749	kerberos-adm
73	rsync	760	krbupdate
79	finger	901	swat
80	www/socks	1243	subseven
81	tproxy/kamanda	6667	irc
87	link	44333	winroute
88	kerberos		

Diese hier aufgeführten Ports sind die Bekanntesten. Sie werden auch als „Well-Known-Ports“ bezeichnet. Andere Nummer können für verschiedene Programme verwendet werden und sind somit dynamisch zu betrachten. Eine Liste mit allen Ports findet ihr auf Happy-Security.de im Tutorial-Bereich. Zum Beispiel könnte ein Trojaner einen dieser dynamischen Ports zum Datenaustausch verwenden, was jedoch meist von der Firewall protokolliert bzw. sogar blockiert wird.
Um nun über das Internet oder Netzwerk auf einen bestimmten Port von einem Rechner zuzugreifen, muss man die IP-Adresse mit entsprechendem Port angeben.
Will man zum Beispiel auf FTP von rhino.acme.com zugreifen, so muss man **ftp 102.54.94.97:21** in der Console eintippen und man nimmt Verbindung zum FTP-Server von rhino.acme.com auf. Verwaltungsprogramme wie zum Beispiel Outlook-Express bedienen sich auch dieser Technik. Bei der Konfiguration eines Mailkontos gibt man die Web-Adresse an, die von dem Programm umgewandelt wird. Hinzu kommt noch, dass man angeben muss über welches Protokoll der Server die Daten sendet und empfängt. (POP3 und SMTP).

Domain Name Service

Da das Internet nun immer mehr genutzt wird und es immer mehr websites gibt, spielt der Einsatz von DNS eine sehr große Rolle. Ohne diesen Dienst, müsste jeder User im Netz wenn er eine Website besuchen will die IP-Adresse angeben... zum Beispiel: 195.135.220.3

Das wäre doch ziemlich anstrengend, oder? Nicht nur das ständige Ziffern tippen und kontrollieren. Man müsste sich die auch noch merken. Aber zum Glück gibt es den Namensdienst DNS. Dieser Dienst befindet sich im Internet und verwaltet jede Internet-Domain mit Namen und IP-Adresse. Gibt man jetzt also www.suse.de in den Browser ein, so schickt der benutzte Browser eine Anfrage nach der IP der Website an einen DNS-Server. Ist dieser Erreichbar, sendet er eine Antwort, anderenfalls wird der nächste Server aufgesucht, der diese Anfrage beantworten kann. Fürs Internet steht der DNS-Server bereit. In einem Netzwerk / Intranet, jedoch können euch eigene „DNS-Server“ aufgebaut werden. Das geschieht alles in einer kleinen Text-Datei, die sich bei windows unter %system32%\drivers\etc\hosts befindet.

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Dies ist eine HOSTS-Beispieldatei, die von Microsoft TCP/IP
# für Windows 2000 verwendet wird.
#
# Diese Datei enthält die Zuordnungen der IP-Adressen zu Hostnamen.
# Jeder Eintrag muss in einer eigenen Zeile stehen. Die IP-
# Adresse sollte in der ersten Spalte gefolgt vom zugehörigen
# Hostnamen stehen.
# Die IP-Adresse und der Hostname müssen durch mindestens ein
# Leerzeichen getrennt sein.
#
# Zusätzliche Kommentare (so wie in dieser Datei) können in
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,
# aber müssen mit dem Zeichen '#' eingegeben werden.
#
# Zum Beispiel:
#
#      102.54.94.97     rhino.acme.com          # Quellserver
#       38.25.63.10     x.acme.com              # x-Clienthost

127.0.0.1       localhost
```

Man kann nun also einfach hingehen und einen neuen Eintag hinzufügen. So zum Beispiel 195.135.220.3 SuSE (Ok, es gibt kein Sinn, aber zum Testen *grins*). Nach dem Speichern kann man nun in der Console mit **ping SuSE** die Adresse 195.135.220.3 anpingen und muss nicht immer die

IP eintippen. Natürlich nur empfehlenswert, wenn man Rechner im Netzwerk hat, bei denen sich das Eintragen lohnt. Von der Syntax: Es muss nur eine gültige IP gefolgt von einem Leerzeichen und dem Zuordnungswort. Es ist eigentlich so, als wenn man eine Variable in einer config-datei deklariert.

In Unix-Systemen heißt die Datei ebenfalss „hosts“, ist jedoch im Verzeichnis /etc zu finden. In diesen Ordnern befinden sich noch ein paar mehr Dateien, in welchen man die Funktionalität für seinen eigenen Nameserver DNS nachahmen kann.

Fake-Mail

Gehen wir jetzt mal wieder einen Schritt zurück zu dem Programm Telnet. Hat man jetzt aber anstatt Port 23 den Port 25 angegeben, so erscheint dort eine Willkommensmessage mit Datum/Uhrzeit des Servers und dessen Version. Wir sind nun also mit dem Rechner verbunden und geben die folgenden Befehle ein:

```
"HELO test.microsoft.com"
"MAIL From:SuSE@home.com"
"RCPT To:gates@microsoft.com"
"DATA"
```

Nun kann die Nachricht eingegeben werden. Es kann auch mit Enter eine neue Zeichen geschaltet werden. Um die Nachricht abzuschließen muss einmal Enter,".",Enter benutzt werden. Wenn soweit geschehen, muss nun "250 Mail accepted" als Antwort vom Server kommen. Oft wurde jetzt die Mail bereits verschickt... Doch schadet es nicht noch mal die Anweisung für die Versendung zu geben. Dies wird mit dem Befehl "SEND" ausgeführt. Somit ist nun eine weitere E-Mail im Umlauf. Das Gute, oder besser gesagt das Schlechte, ist, dass viele Server keine anonyme Versendung von Mails mehr zu lassen. Für noch funktionierende Server einfach mal bei Google suchen.

Das sollte als kurze Einführung in die Möglichkeiten reichen. Da dies eigentlich nur dumme Spielereien sind... beziehungsweise oft dazu missbraucht werden, werde ich mich nun wieder wichtigeren Dingen zu wenden.

Passwortschutz

Der wichtigste Schutz bei den oben erwähnten Diensten um seine Daten zu schützen, ist immer die benötigte Authentisierung durch korrekten Benutzernamen + Passwort! Ohne einen registrierten Benutzernamen inkl. dem richtigem Passwort erhält man keinen Zugriff auf das System. Die einzigste Möglichkeit jedoch wäre noch über das Gastkonto reinzugehen. Diese Kontoart wurde aber extra von den Administrator mit minimalen Rechten versehen, so dass es zumindest gewährleistet ist sich ein wenig im System umzuschauen. Der Administrator, im Gegenzug zu den Gästen und normal registrierten Usern, hat die kompletten Zugriffsrechte und ist sogar in der Lage in den Daten der registrierten User herumzuschnüffeln (auch wenn er es gesetzlich nicht darf, so kommt es dennoch sehr oft vor).

Das gute an diesen eingeschränken Gastkonten ist, dass man keine Spuren ausser vielleicht die IP hinterlässt und dass es trotz der eingestellten Rechte manchmal immer noch möglich ist, an sein Ziel zu gelangen. Da bei einem Angriff über das Gast-Konto immer noch die IP und die Zugriffszeit protokolliert wird, muss man natürlich einen Proxy zwischensetzen, der die IP des Hackers verschleiert. Dies wäre dann also die erste Vorbereitung, die wir treffen müssen, bevor wir uns das System näher anschauen.

Hier jedoch noch mal ein paar Hinweise die bei der Auswahl der Passwörter berücksichtigt werden sollten. Jeder User ist im Normalfall für seinen Account selbst verantwortlich und muss sehen, dass er diese auch ausreichend durch ein Passwort schützt. Der Bequemlichkeitshalber, sind die meisten Menschen unheimlich Faul, was das Ausdenken von besonders originellen Passwörtern angeht. Zugegeben: Es ist ja auch nicht ganz so einfach, aber sollte man nicht zu achtlos dabei sein. Viele verwenden einfach ihren echten Vor/Zunamen oder den Namen des Haustieres. Für einen Hacker ist es in diesem Fall jedoch nicht schwer solch ein Passwort zu knacken, da er durch vorheriges Footprinting (Sammeln von Informationen über das Opfer) schon viele persönliche Daten zusammen hat, die er verwenden kann. Mittlerweile gibt es viele Programme, in denen nur kurz die zusammengetragenen Informationen eingetragen werden müssen. Nach kurzer Rechenarbeit, gibt der Computer nun alle möglichen Kombinationen in einer Liste aus. So hat man eine individuell zusammengestellte Word-List und kann mit einem anderen Programm jede mögliche Kombination ausprobieren. So wären also auch Verschachtelungen wie zum Beispiel: Mario1973 kein schwer zu erratenes Passwort mehr.

Am besten ist es, wenn man sich ein Passwort ausdenkt, das keinen Zusammenhang mit der Person ergibt und dazu auch noch Alphanumerich ist. Ein optimales Passwort wäre dann zum Beispiel: „R3Pc64_x“

1.Aphanumerisch (Buchstaben und Zahlen gemischt)
2.Sonderzeichen (!“§$%&/()=?`°^_)
3.mindestens 8 Zeichen
4.unspezifisch

Wenn man also sein Passwort mit diesen Angaben erstellt, so kann man sich schon mal ziemlich sicher fühlen. Wichtig ist aber trotzdem, dass man überall ein anderes Passwort verwendet. Würde nämlich irgendwo das Passwort herausgefunden werden (gecrackt, keylogger oder ähnliches), so hätte der Cracker zu jedem anderen mit diesem Passwort geschützten Bereich ebenso leicht Zugriff.

In diversen Unix-Systemen kann zusätzlich eingestellt werden, wie lange ein Passwort nur Gültigkeit besitzt. Ist die Gültigkeit abgelaufen, so wird der User aufgefordert beim Einloggen ein neues Passwort auszuwählen bzw. zubeantragen. Dies gibt zusätzliche Sicherheit, falls ein Hacker das alte Passwort bereits hat.

Proxy-Server

Es wäre idiotisch anzunehmen, dass ein Angreifer ohne jegliche Vorbereitungen und Absicherungen einen Server oder eine Website angreift. Zu einer der Vorbereitungen gehört es, einen geeigneten Proxy (oder mehrere) zu finden, über welchem man den Server angreift.

Unter einem Proxy versteht man einen anderen Computer im Internet, über den man sich zum gewünschten Ziel bringen lässt. Man kann sich das ganze wie eine Kette denken:

Normalerweise:

```
  "Hacker"     ======>     "Opfer"
80.135.53.37  ======>  80.135.53.37
```

Wenn das Opfer jetzt aber eine Firewall hat und mitbekommt, dass jemand in den Rechner eindringt, protokolliert die Firewall die IP des Hackers (80.135.53.37). Mit dieser IP kann das Opfer sich beim Provider beschweren und sogar Anklage erheben.

Mit Proxy:

```
  "Hacker"     ------>    "Proxy"    ======>    "Opfer"
80.135.53.37  ------> 137.15.51.133 ======> 137.15.51.133
```

Bei unserem Beispiel versucht der Hacker über einen Proxy den Zugriff zum Rechner des Opfers zu erhalten. Die IP des Hackers ist wieder 80.135.53.37! Er sendet die Anfrage zum Proxy und dieser sendet die Anfrage weiter zum Opfer. Dabei sieht die Firewall des Opfers jedoch nur, dass der Rechner mit der IP 137.15.51.133 (unser Proxy) die Anfrage stellt. Der Rechner des Opfers sendet eine Rückantwort an den

Proxy (klar, da nur die beiden sich unterhalten) und unser Proxy leitet diese weiter an uns. So in etwa funtioniert die Arbeit mit einem Proxy. Da der Proxy ja die Anfrage stellt, werden auch noch alle andere Informationen als nur die IP übermittelt. Eine Liste von aktuellen Proxys kann man unter Socksproxy.de abrufen (natürlich auch bei google zu finden). Leider ist es schwer einen noch funktierenden Proxy zu finden, da die meisten schnell wieder verschwinden und ihre IP wechseln. Unserem Rechner sagen, dass er einen Proxy benutzen soll, ist im Gegensatz dazu ein leichtes Unterfangen. Am besten ist es ein Web-Tool wie Proxomitron zu verwenden. Anleitungen zu Proxomitron sowie zu Proxys allgemein gibt es unzählige im Netz.

Das Problem bei einem Proxy ist immer noch das Restrisiko... es kann immer noch von dem Proxy-Server die Log-Datei durchsucht werden, um die wirkliche IP des Angreifers (80.135.53.37) zu bekommen. Also reicht ein Proxy auch nicht aus um sich wirklich anonym im Netz zu bewegen... es sollte aber vorerst als Schutz reichen. Was noch in die Grundausrüstung eines jeden Netzwerkjunkies gehört ist: IP/Port/Vulnar-Scanner, Remote-Software, sonstige Network-Tools, sowie haufenweise Link-Listen (siehe hierzu: www.happy-security/?modul=links

Usabilities & Toys

Auch wenn immer gesagt wird, Hacker schreiben ihre Tools alle selbst, so kann man nicht einfach die Tatsache ignorieren dass jeder einmal klein angefangen hat und eigentlich viele der tools so wie sie sind schon nahezu perfekt sind (also nicht noch einmal selbst gecodet werden müssen). Natürlich ist es immer gut sein Programme selbst zu schreiben, denn so lernt man am besten, aber wozu das Rad noch einmal erfinden, wenn es schon bereits erfunden wurde und dazu noch immer weiter entwickelt wird? Wäre doch ziemlich dumm. Das erste Werkzeug, das wir benutzen wollen, ist um einen Rechner im Netzwerk ausfindig zu machen. Sehr gut ist, dass dieses usability bereits im System integriert ist (so wie viele andere eigentlich auch). Dies gilt für Win-OS, sowie für alle 'nix Versionen...

PING <<Rechner-IP>> <<Parameter>>
Mit dem Ping-Befehl ist es möglich herauszufinden, ob ein bestimmter Rechner im Internet oder Netzwerk gerade erreichbar ist. Bei diesem kleinen Befehl sendet der lokale Rechner ein Paket, das 32 Byte groß ist, an einen am Netz angeschlossenen Rechner. Kommt das Paket an der Firewall des Rechners an, so wird erstmal überprüft ob dieses Paket in ordnung ist. Nach der Annahme sendet der entfernte Rechner eine Bescheinigung der Annahme zurück an unser lokales System (SYN/ACK-Paket). Jetzt wissen wir genau, dass unser der am Anfang des Befehls eingegebenen Rechner-IP ein Rechner angemeldet ist und empfangsbereit ist. Als Zusatzoption wird mittlerweile für domain-adressen die IP sogar schon umgewandelt. Es ist also auch Möglich die Domain anstelle

der Rechner-IP zu benutzen. Ohne die korrekte IP könnte man keinen Rechner finden. Früher gab es vermehrt die Möglichkeit durch einfache Parameter Abänderung im Ping-Befehl einen entfernten Rechner zum Absturz zu bringen. Es musste nämlich nur die Größe der Pakete etwas hochgeschraubt werden. Mittlerweile wurde das Problem jedoch gefixt und so ist die maximale Paket-Größe auf 65500 Byte beschränkt. Somit ist die Zeit der "PoD-Attacken" endlich fast vorbei (PoD = Ping of Death).

NETSTAT -a

Auch dies ist ein sehr wichtiger Befehl beim Arbeiten im Netzwerk. Mit NETSTAT lassen sich alle aktuellen Verbindungen, die über unseren Rechner laufen anzeigen. Man kann das ganze einfach mal testen, indem man eine Website öffnet und dann den Befehl in der console eingibt. Nun wird eine kleine Liste ausgegeben.

Aktive Verbindungen

Proto	Lokale Adresse	Remoteadresse	Status
TCP	hca-server:http	localhost:1076	HERGESTELLT
TCP	hca-server:http	localhost:1078	HERGESTELLT
TCP	hca-server:1028	localhost:3306	HERGESTELLT
TCP	hca-server:1076	localhost:http	HERGESTELLT
TCP	hca-server:1077	localhost:3306	WARTEND
TCP	hca-server:1078	localhost:http	HERGESTELLT
TCP	hca-server:3306	localhost:1028	HERGESTELLT

Die hier dargestellte Liste basiert auf Testbasis. Bin gerade nicht im Internet, sondern habe es über meinen Lokalen Webserver laufen lassen. Auf jeden Fall für unser Beispiel ok. Es kann gesehen werden, welche Rechner miteinander kommunizieren. Hier sind es lediglich der HCA-SERVER und unser Rechner LOCALHOST. Hinter den Namen und nach dem ":" befindet sich die Angabe des Ports auf welchem die Unterhaltung stattgefunden hat. Der Status gibt den aktuellen Zustand der Sitzung an. In der realen Umgebung sieht das ganze etwas größer aus und ist schwerer nachzuvollziehen. Wenn man aber öfters mal Kontrolliert und weiß, welche Verbindungen offen sein sollten, der wird einen Eindringling schnell erkennen. Oft passiert es auch, dass Microsoft™ versucht nach Hause zu telephonieren um persönliche Daten zu übertragen. Wenn man sich jetzt genauer interessiert, was sich hinter jeder einzelnen Verbindung verbirgt, so kann man einen Sniffer benutzen. Ein Sniffer ist ein Programm, welches die oben angezeigten Pakete protokolliert und ausgibt (meist mit detailierten Angaben). Näheres zu Sniffern und die Anwendung gibt es weiter unter in etwas späteren Kapiteln. Auch noch eine schöne Zusatzfunktion von netstat ist der parameter **-s**. Lässt man sich die Ergebnisse von netstat -s ausgeben, so erhält man eine Liste mit gesendeten und empfangenden Paketen, Datagramen, Fragmenten und Fehlerprotokollierungen.

Zwei andere bereits im System enthaltene Dienste sind **TRACERT und NSLOOKUP.** Beide Dienste geben wieder einmal nähere Informationen über den Server. Der erste Befehl (TRACERT), sendet ein Paket an den Empfänger und protokoliert dabei jeden Rechner im Internet, der dieser Paket annimmt und weiterleitet. Somit kann geschaut werden, über welche Rechner alles läuft und ob ein Proxyserver zwischengeschaltet wurde. Außerdem kann man gut sehen, wielange das Paket von einer Station zur nächsten benötigt. Am Besten ist, wenn man einfach ein wenig mit diesen ganzen Diensten rumspielt und versucht die Informationen zu analysieren. Beim einfachen ausführen dieser Dienste kann noch kein Schaden am Victim-Rechner entstehen.

Wie ihr vielleicht bereits mitbekommen habt, bin ich ziemlich vom eigentlichen Thema abgewichen. Es ging zu anfang ja um Passwortschutz und nun sind wir bei Netzwerkdiensten und Protokollen. Der Grund hierfür ist der relativ nahe Zusammenhang zwischen diesen beiden Themen. Ein System kann einfach nicht sicher sein, solange es nicht mit einem Passwort geschützt ist. Den Entwicklern von Windows ist es anscheind egal, ob ihre User sicher sind. Zu sehen ist dies bereits bei der Installation von Windows und beim Anlegen neuer Benutzer in der Systemverwaltung. Windows überprüft beim Registrieren nicht, ob überhaupt ein Passwort eingegeben wurde oder nicht. Gut, es werden zwar immer wieder neue Sicherheitsupdates herausgegeben, aber diese bringen nichts, solange man sein eigenes System nicht mit einem Passwort schützt. Ok... gehen wir jedoch jetzt einfach einmal davon aus, dass unser Opfer sein System mit einem Passwort geschützt hat. Durch die oben erwähnten Dienste und Protokolle gibt es jedoch immer noch eine Chance auf das System zu gelangen. Entweder durch Hintertüren in Programmen, die ihre Programmierer einmal eingebaut haben oder ganz einfach durch Fehler in den einzelnen Programmen, die es einem Erlauben Dinge mit diesem fehlerhaften Programm anzustellen, die eigentlich nicht damit vorgesehen waren. Solche Fehler in Programmen oder Protokollen nennt man auch Exploits (hatte ich schon einmal weiter oben erwähnt).

TELNET

Obwohl ich eigentlich nicht vor hatte dieses große Tool vorzustellen, tue ich dies auf Wunsch von einigen Kollegen doch noch einmal genauer.

Gestartet wird Telnet wieder über die DOS-Console und zwar mit der Syntax: *telnet <hostname> <port>*

<hostname> ist der Zielrechner, den ihr anpeilen wollt.
<port> hängt von dem Dienst ab, den ihr benutzen wollt.
(bei SMTP=25; FINGER=79;...) Ist der Port auf dem Rechner offen, sprich läuft das Programm hierzu und nimmt eure Anfrage an, so werdet ihr mit dem Rechner (und dem Programm) verbunden. SMTP und FTP habe ich bereits vorgestellt, wie diese funktionieren. Nun noch ein paar

spezielle:

Echo & Charge: Diese Dienste laufen über Port 7 und sind eigentlich nur ein Test, ob der Server alles verarbeiten kann, was ankommt. Unterschied der beiden Dienste ist, dass bei Echo jeder eingegebene Zeichen verarbeitet wird, bei charge werden variable Zeichen gesendet.

Finger: Ein sehr nützlicher Dienst der über Port 79 eine Rückmeldung gibt, welcher User gerade auf der Maschine an dem anderen Ende der Leitung eingeloggt ist.

Nun das Wichtigste: **Über TELNET auf fremden Rechner einloggen.**
Denn hat man sich erst einmal auf einem Rechner eingeloggt und wurde mit genügend Rechten beglückt, so gibt es viele Möglichkeiten, die man anstellen kann (Programme ausführen, Dateien hoch- oder runterladen oder einfach anschauen und ändern).
Hat man sich also mit *telnet <hostname> 23* mit einem Rechner verbunden, so wird nach einem Username und Password gefragt. Meistens kommt man auch als guest in das System. Dann sind wir drinne und können uns im System so fortbewegen, wie früher in DOS... mit Kommandos (ist es ein Linux-System, so müssen die Unix-Commands verwendet werden). Hier nun eine kleine Auswahl von Befehlen.

DOS	UNIX	Auswirkung
DIR	LS	Inhalt von Verzeichnissen anzeigen
CD		Verzeichnisse wechseln
CAT		Inhalt einer Datei ausgeben (ascii)
DEL	RM	Löschen
MKDIR		Verzeichnis anlegen
GET		Datei downloaden
HELP / ?		Listet alle möglichen Befehle auf

Außerdem können mit Hilfe des Parameters -? am Ende eines Befehls ausprobiert werden, was der jeweilige Befehl bewirkt.

Sinn und Zweck einer solchen Aktion für Hacker und Cracker ist es:
1.Passwörter / Accounts zu bekommen
2.Dateien zu stehlen
3.Dateien zu zerstören
4.Server remoten

Am Besten ist es, wenn ihr euch ransetzt und versucht mit den verschiedenen Diensten connected. Einfach mal schauen, was passiert und gegebenenfalls die manpages zur Hilfe nehmen. Wichtig: Wenn Hacker auf fremde Server einhacken und Zugriff erhalten, wird meist die Log-File gelöscht. Es ist also darauf zu achten, dass ein Backup zur Log-File gemacht wird. Außerdem ist es ratsam, falls Dateien erstellt oder verändert wurden, zu schauen, von wem und wann dies geschehen ist.

Was genau sind Exploits?

Ein Exploit bezeichnet ein Programm, oder ein Fragment, welches Schwachstellen in Software oder Hardware ausnutzt. Durch diese Schwachstellen bekommt der Angreifer, der das Exploit benutzt, entweder die Möglichkeit höhere Rechte im System zu bekommen oder bringt einen Rechner zum Absturz (Buffer-Overflow).

Fast jedes entwickelte Programm hat Fehler, die es zum Abstürzen bringt. Nach und nach werden diese Fehler von Anwendern oder Beta-Testern gefunden und an die Entwickler gesendet, damit der Fehler in der nächsten Version ausgebessert werden kann. Es gibt sogar extra Websites und Foren, die sich mit diesem Thema beschäftigen und Fehler auf diesen Seiten veröffentlichen. Für aktuelle Adressen: „bugtraq" in Google suchen. Wird also ein solcher Fehler veröffentlicht, so dauert es nicht lange bis jemand ein Exploit hierzu geschrieben hat, der diesen Fehler ausnutzt. Ein Beipiel wäre hier immer wieder das „phpBB". Dieses Board-System wird immer weiterentwickelt und doch tauchen ständig schwerwiegende Fehler in der Programmierung auf, die es ermöglichen Administrator-Rechte zu erlangen. Es wurde zwar gesagt, das ab der Version 2.0.0 schluß mit solchen Fehlern sei, aber für die neue V. 2.0.8 wurde bereits ein Exploit herausgebracht.

Ein noch viel offeneres Angriffsziel ist der **Microsoft™ Internet Information Server (IIS).** Da immer mehr Administratoren wegen ihrer Bequemlichekeit Microsoft™ NT Server verwenden, haben sich auch dort die Hacker etwas mehr umgeschaut und unzählige Fehler in den Systemen gefunden. Nach kurzer Zeit war auch dort schnell ein neues potenzielles Ziel für Angriffe gefunden. Hierbei konnte man mal wieder schön wie schnell und einfach so ein Exploit entstehen kann. Durch das Herumspielen mit dem Server ist einigen Freaks aufgefallen, dass bei Eingabe von Dateinamen mit einer Länge von über 3000 Zeichen der Server gecrashed wird. Doch damit nicht genug... dies ist ja an sich nur destruktiv, was Hackern nicht viel bringt! Man wollte sich ja die Kontrolle über das System verschaffen. Da über den IIS auch die Windows Shell über Internet erreichbar ist, konnten eine Internet-Security-Crews durch gezieltes Angreifen der Shell Befehle ausführen, die das Anlegen von Super Usern ermöglichte.

Da solche Exploits schon komplett fertig sind und eigentlich nur noch ausgeführt werden müssen, ist es selbst für einen Laien (oder Script Kiddie) nicht schwer, diesen zu benutzen. Oft sind diese Programme in den Händen von Script Kiddies ein viel gefährlicheres Werkzeug, da sie nicht wissen, was genau da passiert und was zerstört werden kann. Für Administratoren und User ist es deshalb umso wichtiger, immer die aktuellsten Version von ihrer WebSoftware mit den neuesten Patches zu besitzen. Leider sind einige (oder besser viele) Administratoren sehr faul, was solche Dinge wie Updates angeht. Deshalb können oft im Internet noch Server gefunden werden, die selbst mit uralten Exploit

noch unter Kontrolle gebracht werden können. WICHTIG: Wenn ihr irgendwo eine Website hosten lassen wollt, dann schaut nach, ob der Server so aktuell wie möglich ist (und am besten auf Linux Basis fungiert).

Da ich den wissendurstigen Lesern hier nun nix vorenthalten möchte, werde ich einmal versuchen zu erklären, auf welche Art sich Lücken bei IIS 5.0 ausnutzen lassen und wie diese geschlossen werden können. Da mittlerweile die Version 6.0 erschienen ist von Microsoft™, rate ich zu allererst natürlich zu einem Update. Wenn dies nicht möglich ist, muss selbst Hand angelegt werden. Kommen wir jetzt aber erstmal zum Angriffsszenario.

Angenommen, wie habe über eine Fehler-Meldung wie zum Beispiel beim Abruf einer Datei vom Webserver: *404-Document not found* . Dort kann man einfach nachlesen, um was für einen Server es sich handelt. Eine andere Möglichkeit wäre, wie oben im Kapitel HTTP-GET beschrieben. Ich erinnere noch einmal daran, dass wir das ganze aus der Sicht des Eindringlings sehen müssen und deshalb auch die Aktionen einmal zumindest im Kopf nachstellen. Wir befinden uns nun auf der Webseite und sehen oben die URL http://iis-suckz.info/docs/index.shtml von hier aus kann man nun ohne viel Aufwand in andere Verzeichnisse wechseln, die sich höher im Baum befinden, als wir eigentlich Zugriff haben. Wie im normalen DOS, kann man mit ../ ein Verzeichnis nach oben wechseln. So kann man von docs/../index.shtml den index von dem stamm-verzeichnis erhalten. Rein theoretisch! So etwas wurde jedoch mit de Version 5.0 von MS gefixed und funktioniert nicht mehr. Da die Umwandlung von Unicode-Zeichen (%20 und so weiter) erst nach der Kontrolle von ../ stattfindet, kann man auch einfach die Äquivaläntes Codes nehmen. In der URL hätten wir dann also statt *docs/../index.shtml* jetzt *docs/..%c0%afindex.shtml* .
Erklärung: %c0%af ist das Unicode-Äquivalent zu / und wird erst noch der Überprüfung der Zeichenkette ../ umgewandelt. Auf diesem Wege kann man nun noch weiter gehen zur Konsole und dort Programme ausführen.
http://iis-sucks.info/docs/*..%c0%af..%c0%afwinnt/system32/cmd.exe*

Hier ein paar websites, die sich mit Exploits beschäftigen. Man sollte hier öfters mal vorbei schauen und gucken, ob wieder ein schwerwiegender Fehler in einem Programm entdeckt wurde.

www.bugtraq.de
www.hack.co.za
packetstorm.securify.com
neworder.box.sk
www.ntsecurity.net
www.eeye.com

TCP/IP-Angriffe

Am Anfang wurd ja nun bereits einmal ausgiebig über den Aufbau von TCP/IP geschrieben. Da diese DAS Grundgerüst vom Internet darstellen, muss ich doch nicht noch einmal erwähnen. Die Protokolle der einzelnen Schichten regeln den gesamten Internetverkehr und müssen aus diesem Grund nahezu perfekt arbeiten. Leider war es zu Beginn des Internets und deren Verbreitung nicht so gut ausgebaut. Viele Hacker haben durch ihre Erfahrungen im Bereich Netzwerktechnologie einige Techniken entwickelt, die es ermöglicht hatten fremde Server zu crashen. Diese kraftvolle Attacke wird DoS-Attack (Denial of Service) genannt und gibt schon einen Eindruck über das gewünschte Ziel von solchen Angriffen. Es sollte nämlich versucht werden das entfernte Gerät (Computer, TK-Anlage) unzugänglich/unbenutzbar zu machen.

Im Laufe der Jahre sind immer wieder verschiedene Arten dieser Angriffe aufgetaucht. Da wäre zum Beispiel LAND, OOB-Attack, ICMP Storm, SYN-Flooding, Teardrop und der wohl bekannteste PoD (Ping of Death). Mehrere Jahre waren diese Attacken schon bekannt, doch konnte man nichts so schnell gegen diese unternehmen. Da es sich um das Internet handelte, konnte man nicht jedem Internet-Nutzer auf der ganzen Welt sagen, er solle von einem auf den anderen Tag das Betriebssystem wechseln oder zumindest updaten (obwohl das cool wäre). Und da es dann auch immer Leute gab, die keine Ahnung haben, hat es ziemlich lange gedauert bis so ziemlich alle dann sicher vor der einen oder anderen Art dieser Angriffe war.

Erklärung der einzelnen Arten:

Ping of Death (PoD)

Tja damit hat alles angefangen. Zwar schon lange her, aber alte Leute werden noch damit geärgert (Win95-User). Im Grunde wurde nur ein großes Paket per PING Befehl an den zu crashenden Rechner gesendet. Wie bereits bei der näheren Erklärung von PING erwähnt, haben solche Pakete eine Größe von maximal 65.535 Byte (65.507 Bytes + 28 Bytes Header-Information). In Windows95 konnte man mit **Ping -l 65510 Rechnername** ein übergroßes Paket versenden, welches vom Empfänger nicht verarbeiten konnte (BufferOverflow). Als Ergebnis wurde nur ein „BSoD“ angezeigt. Mit Windows98 jedoch wurde der Ping-Befehl auf Maximal 65507 Bytes gesetzt, was das Versenden solcher Killah-Kommandos unterbindet.

SYN-Flooding

Auch über SYN habe ich bereits weiter oben schon etwas geschrieben. Dabei ging es um den bekannten „Drei-Wege-Handshake“! Es ist ein Frage-Antwort-Spielchen. Als erstes wird im Normalfall ein SYN-Paket von Rechner A nach Rechner B gesendet. Ist dieser Rechner vorhanden und kann die Anfrage bearbeiten, sendet B ein SYN/ACK-Paket an Rechner A zurück. Zur nochmaligen Bestätigung sendet Rechner A den erhaltenen ACK-Teil in einem ACK-Paket zurück. Es ist wie beim Einkaufen (kaufmännisch). Erst kommt die Bestellung, dann die Lieferung und als letztes noch die Empfangsbestätigung, das die Lieferung ordnungsgemäß angekommen ist. Dabei wird auch immer die Absender und Empfänger-Adresse mitgesendet... sonst weiß der eine Rechner ja nicht, wo das Paket hin gehen soll. SYN-Flooding tritt auf, wenn jetzt ein Rechner seine IP fälscht und ein Paket sendet. Rechner B versucht zu antworten; wartet jedoch vergeblich auf eine Bestätigung zur Lieferung (ACK-Paket). Nach mehrfachem Ausführen, bricht der Rechner zusammen.

OOB-Attacks

Wieder ein Problem, dass eigentlich nicht hätte auftreten müssen. Durch die fehlerhafte Übersetzung von einzelnen Strings in der Drucker-Implementierung über Port 139 von Windows Rechnern, war es möglich einzelne Geräte zu crashen. Das berühmte Tool WinNuke hat diese Schwachstelle ausgenutzt und so Haufeweise Windows-Rechner geplättet.

ICMP Storm

Wie bei der OOB-Attack geht es hier darum, dass man Paket unter andere IP-Adresse versendet. Jedoch sendet man diese Paket nicht zum Opfer sondern schickt sie an eine Broadcastadresse. Als Absender haben wir diesmal jedoch die IP des Opfers genommen, was zum Ergebnis führt, dass die Broadcastadresse dieses Paket weiter an das Opfer sendet. Würden sich nun mehrere Rechner zusammenschließen und diese Technik verwenden, so würde das Opfer von der Flut an Spam-Paketen förmlich platt gemacht. Zum Glück haben hier die Provider schnell geschaltet und das Problem durch entsprechende Router beseitigt.

LAND

Das Besondere an LAND war eigentlich nur, dass dieser Angriff durch Angabe von identischem Absender sowie Empfänger, eine Art Ping Pong Spiel veranstaltete. Da diese Pakete nun also immer weiter durch die Datenbahn raste, kam es zu sogenannten **Race Conditions**. Eine **Race Condition** ist ein Zugriff auf ein und die selbe Resource. Da jedoch nicht nur gleichzeitig gelesen wurde, sondern auch geschrieben wurde, kam es vor, dass ein Datensatz überschrieben wurde. Durch diese Überschreibung der Daten, kam es zum Kollapse im System und es ist zusammengebrochen.

Mittlerweile gibt es sogar das gegenteilige zum BufferOverflow. Wie man sich denken kann, heißt dieser Begriff BufferUnderflow oder BufferUnderRun. Im Grunde werden hier weniger Daten gesendet als nachgefragt werden, was zur Folge hat, das Datenlöcher entstehen und Pointer falsch gesetzt werden.

Diese oben aufgeführten Angriffe sind alle destruktiver Art. Das heißt, sie versuchen etwas beim Opfer zu zerstören oder zumindest zu sabotieren. Es gibt jedoch auch noch Methoden, mit welchen man Zugriff auf das System erhält und Daten ausspähen kann.

Gegen diese ganzen Angriffe gibt es mehrere Möglichkeiten sich zu schützen, aber dennoch kein Allheilmittel. Das Beste wäre, wenn jeder auf Linux umsteigen würde, da hier solche Fehler nicht so häufig auftreten, wie bei Windows-System... MAC lassen wir mal aussen vor. Außerdem wäre der Einsatz von Firewall zum Sperren bestimmter anfälliger Ports (139 bei OOB-Attacks) zu empfehlen. Will man jedoch bei Windows bleiben und traut sich nicht zu viel an seiner Firewall zu konfigurieren, so sollte zumindest immer die aktuellste Version von Windows mit den neuesten Patches verwendet werden. Es sollte auch öfters mal nachgeschaut werden, welche Ports aktiv sind. Vielleicht hat sich ein Trojaner eingenistet, der über einen bisher noch nie benutzten Port mit seinem Wirt kommuniziert. Auch wenn es dann schon zu spät ist, sollte man vor weiteren Schäden vorsorgen.

Selbstverteidigung

Natürlich darf die Selbstverteidigung nicht zu kurz kommen... Wie sehe es denn aus, wenn sich ein Hacker von irgendwelchen kleinen Script-Kiddies hacken lässt? Hi, Tomo *Grins*

Als Grundausrüstung beim Surfen sollte man immer einen Virenscanner und eine Firewall benutzen... Dies sind ist wohl den meisten klar, doch gehört dies nur zum Minimum. Desweiteren sollte man öfters ein paar Tools durchlaufen lassen, die kontrollieren, ob sich nicht irgendwo ein Dialer oder Trojaner eingenistet hat. Dazu jedoch später mehr... kommen wir erstmal zur Firewall.

Firewall

Bei einer Firewall ist es wichtig, dass sie gut konfiguriert ist. Das heißt: Einfaches Installieren und in Auto-Start packen bringt nicht mehr viel. Man sollte sich genau anschauen, was alles mit der jeweiligen Firewall eingestellt werden kann. Öftmals bleiben viele Ports trotz Benutzung einer Firewall weit offen für Angriffe (so Beispielsweise Port 139, den man eigentlich nicht brauch, trotzdem eine große Sicherheitslücke aufweist). Das Vorurteil, dass kostenlose, Personal Firewalls mies sind, ist völliger quatsch. Zwar hat man nicht so viele Einstellmöglichkeiten, wie bei einer Lizenzierten, aber dennoch reicht es meist für den privaten Gebrauch aus. Solltet ihr jedoch einen Dienst, wie HTTP, FTP-Server bereitstellen, so muss noch einmal geschaut werden welche Sicherheitslöcher in der verwendeten Serversoftware stecken.

Es gibt verschiedene Möglichkeiten solche eingerichteten Firewalls ausser gefecht zu setzen. Hier eine kleine Auflistung von möglichen Angriffen gegen Firewalls:

IP-Spoofing

Bei dieser Art der Attacke wird mit Hilfe einer falschen IP-Nummer dem angegriffenen System eine falsche Identität vorgetäuscht. Die gegenseitige Identifikation zweier kommunizierender Netze (Systeme) erfolgt bei den meisten TCP/IP-Protokollen ausschließlich über die IP-Adresse. Im Internet sind jedoch sehr viele "Hackertools" als Freeware erhältlich, die es ermöglichen, eine falsche IP-Adresse vorzutäuschen.

Datenpakete mit gefakten TCP-Headern

Diese Angriffsmethode ist sehr simple. Der Angreifer schickt einem der Netzserver des zu attackierenden Netzes einen unbekannten Paketheader. Der Server interpretiert diesen Header falsch, und wird so zu unvorhergesehenen Reaktionen verleitet. In Folge dieser Reaktionen ist es dem Angreifer dann möglich in das System einzudringen.

Mißbrauch des Source-Routing

Einem IP-Paket läßt sich die Route, die es nehmen soll, um ans Ziel zu gelangen, vorschreiben, genauso wie die Route, den das Antwortpaket zu nehmen hat. Während der Übertragung besteht die Möglichkeit, die Wegbeschreibung zu manipulieren, so daß nicht der vorgeschriebene, sichere Weg (z.B. über die Firewall) genommen wird, sondern ein oder mehrere unkontrollierte Wege.

Mißbrauch des ICMP-Protokolls

ICMP steht für das Internet-Controll-Message-Protokoll. Es hat die Aufgabe Fehler- und Diagnosefunktionen zu übermitteln. Leider läßt es sich zum Ändern der Routingtabellen mißbrauchen, so daß z.B. nicht geschützte Routen benutzt werden. Oder der Angreifer schleust über diesen Weg gefälschte destination-unreachable-Pakte in eine bestehende Verbindung, um diese zu unterbrechen.

Mißbrauch der Routing-Protokolle

Routing-Protokolle haben die Aufgabe zwei vernetzten Systemen evtl. Routenänderungen mitzuteilen. So ist es möglich mit einer dynamischen Routingtabelle zu arbeiten. Für einen Angreifer ist es aber möglich falsche RIP-Pakete (Route-Information-Protokoll)zu erzeugen, und so die Systeme zu veranlassen, ungewünschte Routen zu nehmen.

Eigentlich hatte ich vor hier noch näher auf Firewalls einzugehen, aber durch die wöchentlichen Updates und Patches, kann man einfach nicht sagen welche FW die Beste ist. Zumal es auch von den Resourcen und Einsatzgebieten eures Computers abhängt. Ich selbst bevorzuge ZoneAlarm, da diese gut zu konfigurieren ist und nicht weiter beim surfen nervt wie viele andere (... McAfee ...). ZoneAlarm ist natürlich als kostenloser Download im Netz verfügbar (einfach mal googlen). Bite bedenkt jedoch, dass es wohl nie ein wirklich absolut sicheres Netz geben wird, solange Leute versuchen, in fremde Computer einzudringen. Im Grunde lässt sich jedoch folgendes Schema für Firewalls ausstellen:

Die allgemeinen Ziele von Firewall-Systemen sind folgende:

- Beweissicherung und Protokollauswertung
- Verbergen der internen Netzstruktur
- Zugangskontrolle auf Netzwerk-/ Daten und Benutzerebene
- Rechteverwaltung
- Vertraulichkeit von Nachrichten
- Kontrolle auf der Anwendungsebene
- Entkoppelung von Diensten
- und natürlich: Alarmierung

Malware-Revenge

Nun zu den gesagten Tools, die euch helfen sollen den PC vor Dialern und trojanischen Pferden zu schützen... Solche Programme gibt es mittlerweile wie Sand am Meer und sollen helfen böse Programme, die Daten mitlesen und persönliche Informationen (wie etwa das Surfverhalten) über das Internet versenden, von dem PC zu entfernen. Leider halten die meisten nicht den versprochenden Erfolg und suchen den PC nur oberflächlich nach Malware ab. Immer wieder kommen auch Meldungen, dass sich in verschiedenen Programmen, die eigentlich gegen Spyware sind, selbst Spyware befand. Leider haben die Programmierer dieser Tools jedoch alle, versteckt, in ihren Nutzungsbedingungen solche Klauseln eingefügt, die besagen, dass Daten über den User gesammelt und verarbeitet werden dürfen. Hier aber jetzt eine Liste mit Programmen, die wirklich helfen:

Programm	***Verwendungszweck***
Ad-Aware	Sucht gründlich nach jeglicher Spyware
0190-Warner	Zeigt an wenn ungültige Verbindung aufgebaut wird
Google-Toolbar	Blockt jegliche Popups und sorgt so für entspannteres Surfen
SpyBot S&D	0190-Dialer, Trojaner, Adware oder Keylogger werden leicht aufgespürt und mit Dummy-Files ersetzt, so dass die Programme meist noch funktionieren
Security Task Manager	erkennt potentiell gefährliche Prozesse, welche den PC überwachen oder langsam machen

Versucht einmal selbst mit diesen Programmen zu arbeiten und ordnungsgemäß anzuwenden. Sollten jedoch im Nachhinein noch Fragen zu den oben aufgeführten Programmen auftauchen, so stellt diese einfach bei uns im Forum unter forum.happy-security.de

Informationsbeschaffung

Das Recherchieren im Internet über irgendein Thema wird einerseits immer einfacher, da immer mehr Leute Daten/Informationen online stellen und doch scheint es ebenso auch immer schwerer zu werden, die richtige Information aus dem ganzen Meer an Daten herauszufischen. Egal was man mittlerweile als Begriff in irgendeine Suchmaschine eintippt... es gibt immer Resultate. Viele User im Web probieren nur einige Minuten in einer Suchmaschine ein ordentliches Ergebnis zu bekommen und geben dann schnell erfolglos auf. Wichtig ist aber, dass man vorher nachdenkt, was für eine Suchmaschine man benötigt. Eine Liste mit vielen guten Search-Engines befindet sich auf www.happy-security.de/?modul=links . Mit den meisten Suchemaschinen lässt sich außerdem viel mehr anfangen als nur nach bestimmten Keywords zu suchen. Zum Beispiel kann man Wörter aus der Suche ausschließen oder nur in Titelthemen von Nachrichten suchen lassen. So erhält man schon ein mit Sicherheit oft um 60 % genaueres Suchergebnis.

Es können folgende Grundüberlegungen für die Suche im Netz schon leicht zum Erfolg führen:

> In welche Kategorie geht das zu Findende?
> Muss es Up 2 date sein?
> Welcher Art sollen Ergebnisse angezeigt werden? Text oder medial?
> Die wichtigsten Stichwörter auflisten
> einige Wörter/Phasen herausfiltern

Ist man sich all dieser aufgeführten Dinge im Klaren, so geht es an das Aussuchen der Datenbank (Suchmaschine). Sucht man zum Beispiel nach Informationen über eine Person, einen Laden oder eine bestimmte Website, so muss man andere Referenzen benutzen. Ich empfehle das Telefonbuch, Usenet und Foren, DENIC-Datenbank und andere Archive. Überall dort findet man persönliche Daten oder zumindest noch mehr Informationen zu der gefragten Person. In der DENIC werden alle Domains gespeichert. Dort kann man auch den Namen des Besitzers und dessen Adresse herausfinden. In archive.org sollte man mal vorbei schauen, wenn man Websites besuchen will, die es nicht mehr gibt. Dort werden alte Websites komplett gespeichert und zum Abruf bereit gestellt.

Sollte man auf der Suche nach andere Informationen sein, so empfiehlt sich eine Metasuchmaschine oder große Suchmaschinen wie zum Beispiel: Google.de, Yahoo.com, Altavista.com, alltheweb.com und fireball. Ist man auf der Seite, so lässt man sich auf jeden Fall erst einmal die „erweiterten Funktionen“ anzeigen. Mit mehr Auswahl lässt sich einfach freier arbeiten!

Für Hacker, Cracker und ähnliche Freaks gibt es da natürlich etwas mehr Auswahl an Underground-Suchmaschinen. Eine der bekanntesten ist astalavista.co.uk Dort hat man zumindest vor einigen Jahren richtig gutes Underground Zeugs, wie Überwachungs-Programme, Serials und MP3s gefunden. Wichtig sind auch die bereits erwähnten Datenbanken von Buglists, Viren und Exploits. Hier ist man direkt an der Quelle, auch wenn man ein wenig durchschauen muss.

Es können natürlich immer mehrere Datenbank Anwendungen miteinander kombiniert werden. Auf jeden Fall empfiehlt sich der Besuch auf unserer Link-Liste für den Bereich „searching“.
www.happy-security.de/?modul=links

Web-Programmierung

Als letztes Thema möchte ich nun noch ein wenig tiefer in den Aufbau von Websites und deren Schutz vor Unbefugten eingehen. Da viele Leute hautzutage eine eigene Homepage haben und dennoch nicht genau wissen, wie Sie das geschafft haben. Oft werden Programme benutzt, die für die Gestaltung der Website zuständig sind und einen großen Teil der Programmierarbeit abnehmen. Die Nachteile dieser schönen Programm sind auf den ersten Blick nicht zu erkennen, doch sollte man sich lieber genauer überlegen, ob man mit diesen Programmen arbeitet oder lieber alles per Hand schreibt. Einer der Nachteile, die einem schnell bewusst werden, ist, dass meist unnötig viele Kommentare / Formatierungen in den Quelltext geschrieben werden, wodurch die Datei fast zu platzen scheint... von der Ladezeit ganz abgesehen. Der aber wohl viel wichtigere Punkt bei der Sache ist, dass Nachlässigkeitsfehler in Durchschnittlich jeder 12 Zeile dazu führen, dass eine Sicherheitslücke entsteht. Oft ist diese zwar nicht verheerend, doch sind immerhin noch rund 30 % dieser Sicherheitslücken ein potenzielles Ziel zum Angriff auf eine Website. Davon jedoch hier erstmal genug... kommen wir zu der Entstehung von HTML, der Grundformatierungssprache des Internets, wie wir es heute kennen und lieben gelernt haben.

Die Voraussetzungen dafür, dass Internet auf jedem Rechner & mit jedem Browser annähernd gleich interpretiert wird, muss global gesteuert werden. Hierfür ist das **W3C (World Wide Web Consortium)** zuständig. Diese Organisation stellt die Regeln für HTML und dessen Varianten auf, führt neue Versionen und Standards ein, kontrolliert die Einhaltung von Regeln und überwacht die Entwicklungen. Am Ende hat jedoch der Browser herstellen noch das letzte Wort, da das W3C nur Vorschläge unterbreiten kann, wie HTML Programme aufgebaut werden sein müssen. Die Umsetzung des Codes in Websites wird im Browser entschieden, weshalb es leider schwer ist eine optimale Website für alle Betriebssysteme und Browser Programme zu schreiben. Informationen zu den Regelungen und dem Prinzip von W3C findet ihr auf www.w3c.org

Zwar gibt es mittlerweile sehr viele Tools zum Erstellen von Websites auf unterschiedlichste Arten, doch anstatt einfach eine Anleitung zu einem dieser Programme zu geben, zeige hier kurz, wie man eine „sehr einfache“ HTML-Seite erstellt. Danach gehe ich weiter über zu Java Script und danach in die richtige Programmiersprache PHP.

HTML (Hyper Text Markup Language)

Mit HTML hat alles angefangen... und es wird auch wohl noch eine Weile so weitergebaut, denn auf HTML baut fast jede Website auf. Es ist eigentlich keine richtige Programmiersprache wie viele immer meinen, sondern viel mehr einfach eine Formatierungssprache, die normalen Ascii-Text zusammen mit Bildern in einem Dokument online zur Verfügung stellt. Um das einfache Prinzip der Gestaltung von HTML-Seiten etwas genauer vor Augen zu führen, nun ein Auszug einer Startseite, die in HTML geschrieben wurde. Der Text in dem Feld muss nur (kopiert) in einen Editor eingefügt und als index.html gespeichert werden. Danach muss man die Datei mit einem Browser öffnen und es wird der, im Dokument enthaltene, Quelltext umgewandelt.

```
<html>
  <head>
   <title>Meine Startseite</title>
  </head>
 <body bgcolor=black>
  <font color=#00ff00><b>Willkommen zu meiner Website</b>
  <p>
  <hr width=200 align=left>
  <p>
  Schön, dass du hier bist!<br>
  Hallo Welt... jetzt eine Tabelle:
  <p>
   <table border=1 bgcolor=yellow>
    <tr><td>1. Zeile 1. Zelle</td><td>1. Zeile 2. Zelle</td></tr>
    <tr><td>2. Zeile 1. Zelle</td><td>2. Zeile 2. Zelle</td></tr>
    <tr><td>3. Zeile 1. Zelle</td><td>3. Zeile 2. Zelle</td></tr>
   </table>
  </body>
</html>
```

Es sieht natürlich nicht wirklich ordentlich aus, so wie es jetzt ist, aber es soll nur zu Anschauungszwecken dienen. Eigentlich handelt es sich bei diesen ganzen Befehlen nur um einfache Leitwörter. So wird

zum Beispiel mit '' der Anfang von **Fett formatierten Text** gestartet, bis '' irgendwo im Quelltext steht. Genauso läuft es mit dem Befehl für die Schriftfarbe: '' sagt dem Browser, dass der nun folgende Text bis zu '' in der angegebenen Farbe (*COLOR*) wieder gegeben werden soll. Es gibt jedoch auch Befehle, die keine '/' zum Beenden benötigen, da diese nur ein Element beanspruchen. In diesem, obigen Bespiel meine ich die Befehle für Zeilenumbruch: '
', Absatz: '<p>' und Trennlinie '<hr>'. Das was in '<hr>' weiter steht, sind Parameter. Diese Parameter sind Hilfen um das Element genauer einzustellen beziehungsweie zu konfigurieren. Die Parameter '*width=200' und 'align=left*' geben an, dass 1. die Trennlinie 200 Pixel weit (*width*) sein soll und dass die Linie vom linken Rand verläuft (*align*). Hier nun alle Befehle und einzelnen Variationen nieder zu schreiben, würde ein unendliches Ausmaß annehmen, aus welchem Grund ich lieber auf die Website: www.selfhtml.de verlinken möchte. Dort gibt es Step-By-Step Anleitungen und jeden Befehl akribisch gut erklärt.

JS / JSCRIPT / Java-Script

Da ich finde, dass HTML nicht wirklich interessant ist, beziehungsweise, jeder sich das selbst genauer anschauen kann, wenn es von Interesse ist, so soll er es tun. Dazu könnte zum Beispiel mein Tutorial im Download-Bereich von Happy-Security verwendet werden.

Java-Script ist eigentlich auch keine schwere Sache, doch werde ich hier etwas genauer drauf eingehen und einige einfache Beispiele zeigen, die veranschaulichen sollen, was man mit JS alles machen kann. Ich nehme dazu einfahc mal das von eben genommene Script und füge 3 Zeilen hinter </title> ein und noch eimal bei '<body' **OnLoad=welcome()** hinzufügen... So dass der Anfang ungefähr so aussehen sollte:

```
<html>
  <head>
   <title>Meine Startseite</title>
 <script language="JavaScript">
  function welcome()
   {
     alert("Hallo World!");
   }
 </script>
  </head>
 <body bgcolor=black OnLoad=welcome()>
  ...
 </body></html>
```

Wenn man nun den Quelltext wieder in eine HTML-Datei speichert, so wird beim Öffnen mit dem Browser eine Meldung kommen, auf welcher „Hallo World!“ steht. Das ist Java Script! Genau genommen ist es eine kleine Funktion, die ein Alert-Fenster, wie man es von Windows bereits kennt, aufpoppen lässt. Nun eine nähere Erklärung zu den Befehlen. Mit *<script language=JavaScript>* sagen wir dem Browser, dass nun JavaScript kommt... Das bedeutet, dass die Befehle nicht einfach als HTML abgearbeitet werden dürfen, sondern als JavaScript ausgeführt werden sollen. In der nächsten Zeile definieren wir eine funktion, die von überall im HTML-Dokument abgerufen werden kann. Die Befehle die nun ab der Klammer { bis zu der Klammer } kommen, gehören alle zur Funktion und werden nur ausgeführt, wenn im Quelltext steht dass er ausgeführt werden soll. In unserem Beispiel wäre das: *OnLoad=welcome (), d*a der Name der Funktion *welcome()* ist. *OnLoad* bedeutet, dass die Funktion beim Laden der HTML-Seite aufgerufen werden soll. Man kann jetzt auch ganz leicht schreiben, dass die Meldung beim Schließen der Seite erscheinen soll. Dazu müssen wir nur *'OnLoad'* mit *'OnUnLoad'* ersetzen. Nun ein kleines Rechenscript:

```
<html>
  <head>
   <title>Meine Startseite</title>
  </head>
 <body>
  <script language="JavaScript">
   function check()
    {
     var first, second, Ergebnis;

     first=parseInt(document.rechnen.first.value);
     second=parseInt(document.rechnen.second.value);
     Ergebnis = first+second;

      if (isNaN(first) || isNaN(second))
       {document.rechnen.result.value=("FEHLER: Bitte nur Zahlen!");}
         else
       {document.rechnen.result.value=(Ergebnis);}
    }
  </script>

  <form name=rechnen>
   <input type=text name=first> +
   <input type=text name=second> =
   <input type=text name=result value=?>
   <input type=button value="ausrechnen" onClick="check()">
  </form>

 </body>
</html>
```

Dieses mal ist es schon etwas komplizierter, da wir jetzt mit einem Formular und Variablen arbeiten müssen. Als erstes solltet ihr das

Script kopieren und ausprobieren, um zu sehen, was genau passiert. Als erstes müssen 2 Zahlen in die ersten beiden Eingabefelder (werden in HTML mit *'input type=text'* programmiert.) eintragen. Das 3 mit dem ? Lassen wir frei, da es uns dort das Ergebnis einschreiben soll. Um nun den JavaScript-Teil zu aktivieren, muss einmal auf den Button geklickt werden (hier wieder in HTML: *'<input type=button>'*). Jetzt werden die eingegebenen Zahlen in JavaScript abgearbeitet. Wir befinden uns jetzt in Zeile 3 der Funktion check(): *„first=parseInt (document.rechnen.first.value);“* Hier wird der Wert (*value*) aus dem ersten Eingabefeld (*first*) genommen und in eine Variable zwischengespeichert. Ich habe die Variable *'first'* genannt. Genau wie auch das erste Eingabefeld, welches in mit *'<input type=text name=first>'* deklariert habe. Nachdem auch die zweite Zahl in eine Variable gespeichert wurde, werden beide Zahlen in der Variablen *Ergebnis* zusammengerechnet.

Da es jedoch sehr oft vorkommt, dass Leute anstatt Zahlen auch mal Buchstaben eingeben, habe ich eine Filterfunktion eingebaut, die schaut, ob die eingegebenen Werte in *'first'* und *'second'* beides Zahlen sind. Dies geschieht durch eine If-Abfrage.

```
if (isNaN(first) || isNaN(second))
 {document.rechnen.result.value=("FEHLER: Bitte nur Zahlen!");}
   else
 {document.rechnen.result.value=(Ergebnis);}
```

Die IF-Abfrage sieht von Grundaufbau immer gleich aus, ist also ganz einfach, findet jedoch in jedem Programm verwendung.

```
if ( Abfrageoption(en) )
 { Funktion die ausgeführt werden sollen wenn Abfrage korrekt }
   else
 { Funktion die ausgeführt werden sollen wenn Abfrage nicht korrekt }
```

In unserem Beispiel wird also in der Abfrage mit *'isNaN(first) || isNaN(second)'* kontrolliert, ob *first* oder *second* einen Fehler geben. Ist dies Korrekt, also trifft es zu, so wird der erste Teil ausgeführt mit den enthaltenen Befehlen. Ist alles gut gegangen und war kein Fehler dabei, so wird der Teil, der nach *else* folgt ausgeführt. Bei uns heißt das im Klartext, entweder (also wenn Fehler ist) die Meldung „FEHLER: Bitte nur Zahlen!“ wird in das dritte Feld ausgegeben oder der zusammengerechnete Wert aus *Ergebnis* wird in das dritte Feld geschrieben.

Schön haben wir auch dieses Beispiel abgehandelt und hoffe es verständlich erklärt zu haben. Falls jedoch noch Fragen hierzu auftauchen sollten, so findet ihr sicher in unserem Forum Hilfe.

PHP – Hypertext Preprocessor

Kommen wir nun zu meinem jetzigen Lieblingsthema PHP. Angefangen habe ich mit HTML Websitengestaltung von ungefähr 5 Jahren und habe am Ende eine Pause eingelegt, da es langweilig wurde immer nur diese einfachen Befehle hintereinander wegzuschreiben. Dann jedoch kam ich auf PHP. Keine Ahnung wie, aber es war auf jeden Fall eine gute Erfahrung für mich, was Webdesign und Programmierung anging und ich kann nur jedem empfehlen auch mal PHP zu lernen. Gründe hierfür sind:

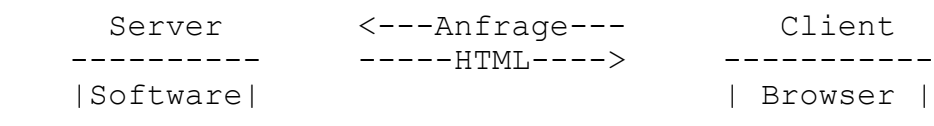
- Es ist einfach
- Man benötigt keine aufwändige Entwicklungsumgebnung
- Syntax von anderen Programmiersprachen; also leichter Umstieg
- gute Ergebnisse bei Webprogrammierungen
- Vereinfacht HTML Webgestaltung durch templates
- fast von grandioser Funktionsumfang
- Plattformunabhängig
- Ziemlich Sicher, da Serverseitig
- gute Nutzbarkeit mit Datenbanken (SQL, Oracle, ...)
- und das Wichtigste: **OPEN-Source**

Dies sind die wichtigsten Vorteile von PHP. Es ist einfach geil. Ok, ok, genug der Schwärmerei; lasset ~~D~~ Taten sprechen.

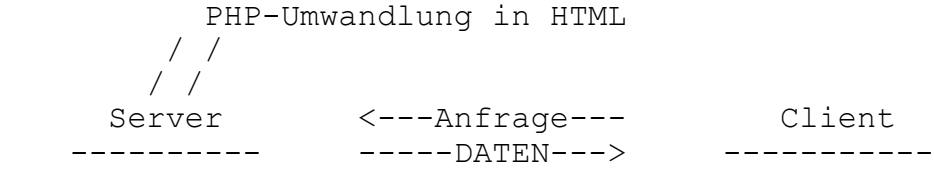
Entstanden ist PHP im Jahre 1995 unter dem Namen PHP / FI (Form Interface) und wurde von Rasmus Lerdorf kreiert. Da PHP jedoch Open-Source ist (anders als andere Programme), wurde es mit rasanter Geschwindigkeit von Hobby-Programmierern und Hackern weiterentwickelt (anders als Mircosoft-Produkte). Aus diesem Grund entschied man sich auch PHP vollkommen Plattformunabhängig zu gestalten und mit einem eigenen Interpreter auszustatten, der auf einzelne Serversysteme aufgespielt wurde.

Da man PHP ganz einfach wie HTML mit dem Editor programmieren kann, also ohne kompilieren oder ähnliches, besteht auch die Möglichkeit PHP-Code in HTML-Dokumente mit einzubinden. Ohne Komplikationen wird der PHP-Code gesondert von HTML-Code abgearbeitet und ausgeführt. Besser gesagt, der PHP-Code wird auf dem Server ausgeführt und als HTML an den Client-Browser gesendet, der den HTML-Code dann komplett in die Website umwandelt.

Normal HTML-Seite (ohne PHP):



PHP-Website (gemischte Daten):



Man kann hoffentlich erkennen, was es darstellen soll. In der ersten Abbildung, wird eine Anfrage vom Client zum Server gesendet. Dort werden Daten einfach ausgetauscht. Beim zweiten Beispiel jedoch, werden nach der Anfrage die PHP-Codeschnipsel im Dokument erstmal im Interpreter auf dem Server ausgeführt und abgearbeitet. Danach wird der entstandene HTML-Code mit dem normalen HTML-Code normal weiter zu Client Browser gesendet, welcher diese dann wieder in eine Website zusammenstellen lässt. Aus diesem Grund kann man auch nur mit PHP arbeiten, wenn auch wirklich eine Verbindung zu einer Website erstellt wird. Wird keine Anfrage zum Server gesendet, so arbeitet das Script auch nicht. Es ist also nicht als Aktive Programmiersprache zu bezeichnen. Mit PHP können Websites dynamisch werden, jedoch bleibt es nicht aus, HTML zu benutzen, da sonst keine Ausgabe im Browser erscheint. Nun das erste Beispiel wie einfach es ist, PHP-Code zu schreiben und in HTML einzubinden:

```
<html>
 <body>
  Dies ist unsere Welt; die Welt der Elektonen und Schaltungen.
<?PHP
 echo 'Der Schönheit des Baud!<p>';
?>
  geschrieben von The_Mentor
 </body>
</html>
```

Bevor jedoch jetzt versucht wird, krampfhaft diese Daten zu speichern und im Browser anzuschauen, noch eine Aufklärung. Auf dem heimischen PC mit PHP zu arbeiten ist in diesem Fall nicht ganz so einfach, wie mit HTML. Mehrere Sachen sind diesmal zu beachten:

1.Als Dateiendung nicht mehr .html , sondern .php nehmen
2.Wir brauchen einen Server, der den PHP-Code interpretieren kann

Das Erste wird ja keine Schwierigkeit sein, doch müssen wir jetzt entweder ins Internet und einen Webhoster suchen, der PHP unterstützt oder wir erstellen uns einfach einen Webserver selbst. Ich bin für die zweite Variante... das ist besser zum Lernen und für spätere Arbeiten sicher noch hilfreich.

Um einen Webserver auf einem PC zu erstellen, müssen wir erstmal ein Tool namens FOXSERV 3.0 herunterladen (Ist auf unserer Seite im Download-Bereich zu finden). Wir gehen doch jetzt mal davon aus, dass ihr alle Windows-User seid, also: mit dem *Microsoft Windows Installer* das heruntergeladene Programm ausführen installieren. Eigentlich sollte alles ausführlich erklärt sein und zumindest ein Hilfe-Dokument .chm mit beiliegen. Es müssen unteranderem Pfad für den Server angegeben werden und ein Account angelegt werden (soweit ich noch weiß). Ist die Installation geglückt, kann der Server mit *'START -> Ausführen -> START APACHE'* in der Console ausgeführt werden. Danach muss noch der Browser geöffnet und die lokale Adresse des PCs eingegeben werden... ja, die hatten wir schon mal: localhost oder 127.0.0.1! Nun sollte eine Website erscheinen, auf der steht, dass es geklappt hat.

Leider kann man die PHP-Datei nicht einfach mit Doppleklick ausführen, sondern muss sie erstmal in den vorhin angelegten Webserver-Ordner speichern und über die URL aufrufen. Wenn die Datei 1st.php heißt, dann müsste man sie mit 'http://localhost/1st.php' im Browser aufrufen können. Nun wieder ein paar Beispiele mit PHP. Ich werde mich jedoch auf kleine Beispiele beschränken, da es sonst zu unübersichtlich wird.

```
<?PHP
                              // wieder ein Rechenbeispiel
 $zahl1 = 23;                 // Variable 1
 $zahl2 = 5;                  // Variable 2
 $ergebnis = $zahl1 - $zahl2; // Variable 3

// Ausgabe verbunden mit einem String und der Variable3
echo 'Die beiden Variablen ergeben zusammen: <b>'. $ergebnis .'</b>';
?>
```

Ich hoffe, dass ich nicht erklären muss, was genau hier passiert. Hier haben wir eine (noch) etwas schlechtere Methode um einen Rechner zu schreiben. Was das Miese hierbei ist? 1. man muss die Zahlen immer im Qualtext ändern und es kann nur Zahl2 von Zahl1 abgezogen werden.

Aus diesem Grund wollen wir jetzt in Verbindung mit HTML wieder ein Formular einfügen, mit dem die Zahlen eingetragen werden können. Als kleinen Zusatz noch ein Feld, das den Operator ändern lässt (*+-/).

```
<?PHP
  $submit =@$_REQUEST[submit];
  $zahl1 = @$_REQUEST[zahl1];
  $zahl2 = @$_REQUEST[zahl2];
  $operator = @$_REQUEST[operator];
if ($submit=="rechnen") {
  if ($operator=="+") { $ergebnis = $zahl1 + $zahl2;}
    elseif ($operator=="-") { $ergebnis = $zahl1 - $zahl2;}
    elseif ($operator=="*") { $ergebnis = $zahl1 * $zahl2;}
    elseif ($operator=="/") { $ergebnis = $zahl1 / $zahl2;}
  else { $fehler = "<font color=red>Es ist ein Fehler aufgetreten!</font>"; }

  if ( @!$fehler) {echo 'Das Ergebnis von '. $zahl1.$operator.$zahl2 .' lautet <b>'. $ergebnis .'</b>';}
  else  {echo $fehler;}
}
?>
<form>
<input type=text name=zahl1 size=10>
<input type=text name=operator maxlength=1 value=+ size=1>
<input type=text name=zahl2 size=10>
<input type=submit name=submit value=rechnen>
</form>
```

Dies ist ein schönes Beispiel um einen Mini-Rechner zu bauen. Wir haben wieder ein Formular, in welchem diesmal 3 Eingabefelder vorhanden sind. Das erste Feld ist Zahl1; das Zweite ist der Operator; und das Dritte ist Zahl2. Der Operator kann nur ein Zeichen aufnehmen, da wir ja auch nur einen Operator zum Rechnen übergeben wollen (+, -, * oder /). Wenn wir also beide Zahlen (zb.: 23 und 5)und einen Operator (+) eintragen und auf den Button klicken, so werden die Daten an unseren Server gesendet und weiter verarbeitet. Die Variablen werden jetzt per URL an den Server gesendet:
http://localhost/2nd.php?zahl1=23&operator=+&zahl2=5&submit=rechnen

Hierbei wird jetzt die Datei noch einmal geladen, doch werden jetzt die Daten, die wir eben in die Eingabefelder eingetragen haben mit übertragen. Die Daten werden dabei ganz einfach an die URL

drangehangen. Der Anfang von der Auflistung der Daten wird mit einem '?' angezeigt. Danach wird der Name des ersten Eingabefeldes gefolgt von dem dazugehörigen Wert drangehangen. Syntax: *'Variablenname=Wert'* also in unserem Beispiel *'zahl1=23'* . Danach kommt das zweite Eingabefeld: *'operator=+'* (diese beiden werden jeweils durch ein '&' getrennt). Ok, ich denke das ist verstanden. Wir haben jetzt die Variablen übermittelt und gehen jetzt mit diesen in den Quelltext. *'$_REQUEST[zahl1];'* hiermit wird die Variable *'zahl1'* in Empfang genommen und kann weiter verarbeitet werden.

Mit einer IF-Abfrage testen wir nun ob wir überhaupt Variablen haben, die verarbeitet werden sollen. Das Problem wäre nämlich, hätten wir keine Variablen übertragen, so gäbe es einen Fehler aus. Danach Testen wir mit, ob der Operator zulässig ist. Würden wir beim Senden einen falschen Operator übermitteln, so könnte nicht richtig gerechnet werden und schon gäbe es wieder eine Fehlermeldung. Dabei wird mit if ($operator=="+") getestet, um welchen Operator es sich genau handelt und führt daraufhin die Rechnung die, die in *'$ergebnis'* gespeichert werden. Sollte es kein gültiger Operator gewesen sein, so wird in die Variable *'$fehler'* eine Fehlermeldung geschrieben. Dies ist in unserem Beispiel noch nicht so erforderlich, da wir ja wissen wo der Fehler liegt, aber wenn es mehrere potenzielle Fehlerquellen gibt, dann ist es schon sehr hilfreich.

Als letztes wird dann noch einmal geschaut, ob die Variable $fehler existiert und gibt dementsprechend die Ausgabe entweder mit der Fehlermeldung oder der Rechnung aus.

Meiner Meinung breche ich hier lieber den kleinen Kurs in PHP ab, da es ein weiteres Buch füllen würde, wenn ich hier noch weitere Bespiele aufführen würde, wie man PHP-Scripte schreibt. Erklärungen wie die Funktionen funktionieren und was mit diesen gemacht werden kann, würde viel zu viel Platz in Anspruch nehmen und sind deshalb noch als einzelnes Tutorial bei uns im Download-Bereich zu finden.

Website-Hacking

Kommen wir jetzt endlich zu dem Thema, das die Meisten unter euch hier interessiert. Wir hacken uns in Websites ein!!! *HaR* *HaR*

Naja gut so wie ihr euch das jetzt vorstellt wird es noch werden. Obwohl es sehr gut für das Grundverständnis ist, bringen diese hier aufgeführten Beispiele wohl keinem einen großen Nutzen. Es wird sicher noch einige Seiten geben, die solche Sicherheitslöcher besitzen, aber diese sind meist Extra eingebaut. Extra? Ja, extra... Die Rede ist von sogenannten Hackits-Seiten. Auf Hackits-Seiten befinden sich viele passwortgeschützte Internet-Seiten, die es geht zu hacken. Hier kann jeder üben zu hacken und von anderen Usern in der Community lernen. Auch bei uns auf Happy-Security.de befinden sich solche Hackits. Schaut einfach mal nach dem Lesen dieses Buches dort vorbei und versucht ein paar Websites zu knacken. Jetzt aber erstmal los:

Einfacher JS-Webseitenschutz

```
<script language="Java Script">
var pw = 'goodvirus', passwort = document.login.pass.value;
if(pw == password) {
alert('geschafft');
}
else
{
alert('sorry falsch');
}
</script>
```

Erklärung: Dies ist jetzt ein kurzer Auszug des JS-Teils. Der HTML-Teil ist nicht wichtig. Es befindet sich einzig uns allein ein Formular in dem HTML-Teil mit einem Passworteingabefeld namens 'pass'. Wenn man sich das Script oben einmal kurz durchgelesen hat, so sollte man das Passwort schon sehen können... auch wenn man keine Ahnung von Programmierung hat. Mit der If-Abfrage wird getestet, ob unser eingebenes Passwort (die Variable 'password') mit dem aus dem Script übereinstimmt (Variable 'pw').

Vor einigen Jahren, bevor der Internetboom los ging, reichte solch ein Passwortschutz aus. Warum? Obwohl man keinen richtigen Schutz hatte. Man sieht alles, aber nur wenn man danach sucht. Wie die Magier es auch immer machen... was man nicht sieht, existiert auch nicht. Früher haben nur wenige Leute mal in den Quelltext geschaut um zu schauen wie die Seite aufgebaut ist oder besser es haben nur wenige Leute gewusst, dass man den Quelltext anschauen kann. Falls dir auch noch nicht

gezeigt wurde, wie du an den Quelltext einer website kommst, so wird dir hier geholfen. Der einfachste, aber mittlerweile nicht immer effektive Weg ist, auf einer Website die rechte Maustaste zu benutzen. Es erscheint ein Menu in welchem etwas von „Quelltext anschauen...“ stehen müsste (je nach Browser unterschiedlich). Am besten du versuchst es einfach mal anhand unserer Website www.happy-security.de/?modul=hacking-zone. Ist die Seite geladen, einfach das eben erklärte machen, danach hat man den Quelltext und muss nur ein wenig suchen.

Sollte es dem Lesen nicht funktionieren, weil es vielleicht etwas gemeiner versteckt ist, so kann man sich den Wert, nach dem gefragt ist (bei uns ist es der Inhalt der Variablen 'pw'), einfach mit einem Befehl ausgeben lassen am Bildschirm. Der Befehl lautet alert() und wurde netterweise schon von mir für die Meldung, ob es geklappt hat oder nicht, eingebaut. Jetzt müssen wir das script nur so umschreiben, das er die Variable ausgibt anstelle der Nachricht: „sorry falsch“. Wichtig ist dabei, dass Text immer in „“ geschrieben werden müssen und Variablen ohne geschrieben werden. Das bedeutet, wir nehmen was in den Klammern steht raus und ersetzen es durch „pw“ (ohne Anführungszeichen ! ! !). Jetzt noch speichern und einmal ausführen, schon bekommt man das Passwort vom PC ausgespuckt. Diese Methode geht bei vielen Hackits, jedoch nicht bei allen und nicht nicht sehr ehrenhaft, da man nicht viel dabei lernt.

Nun eine etwas fiesere Methode:

```
<script language="JavaScript">
function pass()
{
var eingabe=document.code.eingabe.value;
var pw = document.title; pw2 = pw.substring(4,6);
  pw= pw2+pw2;
 if (eingabe == pw) {
  alert("jo das war richtig.");
   } else {
   pw=document.title; alert("wrong password");
 }
}
</script>
```

Erklärung: In diesem Script befinden sich schon ein paar Schwierigkeiten für uns... für den Admin besser gesagt bugfixes. Als erstes wird aber ganz gewohnt unsere Passworteingabe in eine Variable gepeichert (var 'eingabe'). Danach wird das Passwort deklariert und verarbeitet. Hierbei kommt schon die erste Erneuerung vor: Der Inhalt

der Variablen wird mit einer String-Funktion bearbeitet und verändert somit unser Passwort. Zu anfangs war es noch *'pw=document.title'*, was soviel bedeutet wie den Titel der Seite als Passwort zu nehmen. Da nun aber die besagte String-Funktion kommt, die das Passwort bearbeitet, verändert sich das Passwort nochmal. Wenn wir als Titel der Seite zum Beipiel: „0123456789“ nehmen würden, würde durch die String-Funktion substring(); als neues Passwort „45“ herauskommen. Die Funktion substring sorgt dafür, dass nur ein ausgewählter Teil einer Zeichenkette genutzt wird.

SYNTAX: string VARIABLE.substring(X,Y);

Das bedeutet, für VARIABLE nehmen wir den Text, den wir bearbeiten wollen, mit X geben wir das Anfangszeichen an und mit Y das Endzeichen. X und Y sind die Stellen von den gezählt wird. Es sind also Zahlen zu verwenden. Bitte immer bedenken, dass beim Programmieren bei 0 angefangen wird. Da unser Titel „0123456789“ ist, wäre substring(4,6); die Kette vom fünften bis (!) zum siebten Zeichen. Sollten dies noch nicht ganz klar geworden sein, so schaut einfach im internet unter selfhtml.de nach substring (in der Rubrik Java Script).

Jetzt noch eine gemeine Sache. Die meisten denken, dass in der Zeile „pw=pw2+pw2;“ die beiden pw2 einfach zusammen gerechnet werden, aber dem ist nicht so. In anderen Programmiersprachen ist es zwar so, aber in Java Script werden diese Beiden Variablen einfach mit einander verbunden. Das bedeutet in unserem Beipiel mit dem Wert „45“ von 'pw2', dass wir nun „4545“ für 'pw' herausbekommen. Dies wäre jetzt also das Passwort, welches wir benötigen um die Seite zu knacken.

PHP Voting faken

Dies ist eines der neuer Aufgaben die es geht zu bestehen... Man muss ein Vote fälschen oder unbrauchbar machen. Nehmen wir mal an, wir könnten in der Schule mit allen Schülern zusammen beraten, wer welche Schulnote bekommt und es wird entschlossen, das ganze über eine Abstimmung zu machen... nicht mehr per Handzettel, sondern digital (online). Jeder Schüler kann eine Stimme für jeden Schüler abgeben (auch für sich selbst). Für uns würde das heißen, dass wir nur einmal für uns voten können und dabei eine möglichst gute Note erzielen wollen. Normalerweise ist dies ja eine 1. Doch da vermutlich in der Datenbank nur der Wert der bereits eingetragenen Zensur mit dem neuen per Quersumme ermittelt wird, wäre es am besten, wenn wir zum Beispiel eine -2 geben würden. Hier jetzt ein Beispiel: Wir haben zur Zeit eine 4 und wollen auf eine glatte 1 kommen.

Datenbank	***Eingabe***	***Ergebnis***	***Auswertung***
4	1	2,5	Nicht geschafft
4	-1	1,5	Nicht geschafft
4	-2	1	geschafft

Wenn wir jetzt also eine -2 an die Datenbank übertragen könnten, dann würden wir unsere 1 bekommen. Würden wir ein Zahl wie -200 senden, so kämen wir auf eine Zensur von -98... ***lol*** ich bin sicher, dass das am Ende auffallen würde. Sehen wir uns jetzt einmal das Formular im Quelltext an:

```
<html>
 <body>
Herzlich willkomme auf der DAU-Schule.<p> Bitte stimmen Sie hier für den von Ihnen
ausgewählten Schüler ab. Eine 1 wäre 'sehr gut', bis zu einer 4 ist 'ausreichend'
und eine 6 ist 'schlecht'.<p>
<b>Schüler: Jonny Lee Miller</b><br>
 <form>
  <input tyoe=hidden name=class value='12A'>
  <input type=hidden name=forpupil value='Jonny Lee Miller'>
  <input type=hidden name=frompupil value='Jonny Lee Miller'>
  <select name=zensur>
   <option value=1>- 1 -</option>
   <option value=2>- 2 -</option>
   <option value=3>- 3 -</option>
   <option value=4>- 4 -</option>
   <option value=5>- 5 -</option>
   <option value=6>- 6 -</option>
  </select>
  <input type=submit name=abstimmen value=abstimmen>
 </form>
 </body>
</html>
```

An sich ist das ganze nicht wirklich schwer zu ändern... Uns geht es aber darum, es richtig zu verstehen. Also fangen wir wieder ganz am Anfang an... ach ja: wir sind natürlich jetzt 'Jonny Lee Miller' und wollen uns einmal selbst bewerten... Wie schon einige Seiten zuvor im PHP-Kurs erklärt, kann man die Wert, die in einem Formular versendet werden, mit PHP abfangen und verarbeiten. Leider haben wir keinerlei Möglichkeiten, an den Quelltext des PHP-Teils heranzukommen, da dieser auf dem Server liegt und nur dort ausgeführt wird. Sobald wird auf den

Knopf drücken würden, werden die Daten gesendet werden. Im Normalfall senden wir diese ganzen Daten.

- class = 12A
- forpupil = Jonny Lee Miller
- frompupil = Jonny Lee Miller
- zensur = 1
- abstimmen = abstimmen

Das Schöne an der Sache ist, dass diese Daten wieder über die URL gesendet werden. Hieße die Datei, in der dieses Formular arbeitet zum Beispiel 'zensuren.php', dann würde die URL folgend aussehen:
http://www.dau-schule.de/zensuren.php?class=12A&forpupil=Jonny+Lee+Miller&frompupil=Jonny+Lee+Miller&zensur=1&abstimmen=abstimmen

Wenn wir jetzt einfach einen der Werte ändern, hätten wir schon das Formular überwunden und gefaked. Was wir wollten, wäre aus der 1 eine -2 zu machen, so dass wir eine glatte 1 bekommen. Hier wäre es jetzt auch ganz einfach möglich, sich als eine andere Person auszugeben und unter dessen Namen zu voten (der Wert, der in 'frompupil' müsste dabei geändert werden). Oft ist die Option ausgeschaltet, dass die Daten über die URL angehängt werden. Dann kann man aber trotzdem versuchen, selbst so eine URL zu schreiben durch die Namen der Formularkomponenten. Meist klappt es dann trotzdem .^_^.

Kommen wir jetzt aber zu den wirklich lehrreichen Sachen dabei... und zwar dem Beseitigen dieser Sicherheitslücken. Also falls ihr so ein schlecht konfiguriertes Formular findet, nutzt es nicht aus, sondern schreibt den Administrator an und erklärt ihm die schwerwiegende Fehlerquelle.

Das Wichtigste, was es zu fixxen gilt wäre:

1. nur die echten Notenwerte zulassen und gefakte Werte mit einer Fehlermeldung zu protokollieren. In PHP würde das so aussehen:

```
<?PHP
$zensur = $_REQUEST[zensur];
if($zensur<1 || $zensur>6) {
$Fehler = 'Es wurde versucht eine Zensur zu faken';
}
else
{
// Normal weitermachen
```

2. zusätzlich vielleicht noch eine Filter für Buchstaben und Sonderzeichen einbauen.
3. 'frompupil' entfernen und dafür einen Cookie oder einen Login zum testen, ob der echte Schüler am PC sitzt, nehmen.
4. In Datenbank kontrollieren, ob Schüler schon gevotet hat.
5. In Datenbank kontrollieren, ob Schüler überhaupt existiert.

SQL-Injection

Dieses Thema ist seit einiger Zeit in jeder Munde, wenn es um Sicherheit bei Webseite geht. Fast jede Webseite benutzt MySQL-Datenbanken und fast jede ist gegen SQL-Injection anfällig. SQL-Injection bezeichnet das Einschleußen von fremden Code in webseite, wodurch dieser dann an bestehende Datenbankbefehle angefügt oder eingebettet wird. Meistens wird hier, genau wie beim normalen PHP-Hacking, der Code über die URL eingeschleußt. Werden also Eingaben, die beim Übermitteln von Formularen an ein Script, nicht richtig geprüft und maskiert, so gibt es hier die Möglichkeit für einen Angreifer Befehle ausführen zu lassen. Hier jetzt kurz eine Erläuterung, wie man Kontakt zu einer SQL-Datenbank aufnimmt und im späteren Verlauf, wie man dort bösartigen Code einschleußt und die Webseite übernimmt. Angenommen, ein Besucher kommt auf unsere Webseite und will sich einloggen. Standardmäßig wird dabei ganz einfach der **Benutzername** und das dazu gehörige **Passwort** ins Formular eingegeben und über den Submit-Button an den Server gesendet. Dieser wandelt das Passwort in eine Hash um und lässt es weiter zur Datenbankabfrage. Diese sieht ungefähr so aus:

```
$user_result = mysql_query("SELECT user_id, user_group FROM members
WHERE user_name = '".$user."' AND user_pass = '".$hash."' LIMIT 1");
...
```

Uns interessiert jetzt nur diese Zeile dort. Mit diesem „Befehl“ sagen wir, dass wir gerne aus der Tabelle *members* (*FROM members*) den Inhalt der Spalten *user_id* und *user_group* (*SELECT user_id, user_group*) hätten. Bedingung muss jedoch sein, der in *$user* übertragene Benutzername in der Spalte *user_name* der Datenbank vorhanden ist (*WHERE user_name = '“.$user.“')*. Da der Account jedoch noch mit einem Passwort geschützt ist, muss hier noch Kontrolliert werden, ob das eingegebene Passwort mit dem gespeicherten Wert von *user_name* übereinstimmt. Weitere Infos zu md5-Hash in Datenbanken im Thema **Passwort-Hashing** weiter unten.

Angenommen wir haben folgende Tabelle *members*:

```
+user_id+--user_name--+-----------user_pass------------+-user_group-+
|     1 | Karl        |1b992e39dc55f0c79dbe613b3ad02f29|          1 |
|     2 | Hubert      |5a105e8b9d40e1329780d62ea2265d8a|          2 |
|    10 | Hans        |60474c9c10d7142b7508ce7a50acf414|          2 |
|    ...|             |                                |            |
|    23 | King        |16d7a4fca7442dda3ad93c9a726597e4|          0 |
|    24 | Nimda       |098f6bcd4621d373cade4e832627b4f6|          2 |
+-------+-------------------------------------------+------------+
```

Wie der Datenbank mitgeteilt, wollen wir nun *user_id* und *user_group* haben. Wenn sich jetzt also Nimda versucht einzuloggen, muss der übersendete URL so aussehen: /login.php?user=Nimda&pass=test . Aus dem Passwort wird aus Sicherheitsgründen auf dem Server der Hash 098f6bcd4621d373cade4e832627b4f6 generiert und sieht die Datenbankabfrage dann folglich aus:

```
$user_result = mysql_query("SELECT user_id, user_group FROM members
WHERE user_name = 'Nimda' AND user_pass = '098f6bcd4621d373cade4e832627b4f6'
LIMIT 1");
...
```

In dem Fall würden wir also als *user_id* = „24“ und als *user_group* = „2“ erhalten. Soviel zum Standardlesen aus einer Datenbank. Was aber, wenn der Besucher ein wenig rumspielt und statt seines richtigen Passworts einfach mal *ABC* eingibt? ... nichts ... naja gut, das Ergebnis würde 0 Zeilen zurück geben und eine Fehlermeldung ausgeben, dass das Passwort falsch wäre. Dies ist noch recht unspannend. Würden wir nun aber ein wenig mehr Chaos versuchen zu machen und *?user=Nimda&pass='* eingeben, bricht meistens der query ab und erzeugt einen Error mit nützlichen Informationen für Angreifer.

> „You have an error in your SQL syntax. Check the manual that corresponds to your MySQL server version for the right syntax to use near „Nimda“ AND user_pass = '“ at line 23.“

Jetzt hätte Nimda bereits Kenntniss über die Bezeichnung des Feldes, wo das Passwort gespeichert wird und darüber hinaus, dass das Script anfällig gegen Usereingaben ist. Wäre hier jetzt ein Angreifer am Werk, könnte einfach statt des Passwortes ***'or 1='1*** eingegeben werden und schon ist man als gewünschter User im System. Erläuterung:

```
$user_result = mysql_query("SELECT user_id, user_group FROM members
WHERE user_name = 'Nimda' AND user_pass = ''or 1='1' LIMIT 1");
...
```

Wie man hier nun eigentlich ganz einfach lesen kann, wird überprüft, wo der *user_name = Nimda* ist und, ob das user_pass entweder ' ' (also leer) oder ob 1 den Wert 1 hat XD ... Logischerweise ist 1 immer gleich 1. Da kann selbst Bill Gates nix dran rütteln und somit ist das Statement TRUE und *user_id* sowie *user_group* von Nimda werden ausgegeben. Natürlich kann man nun sich auch als Root/Admin anmelden, sofern man weiß, wie der Benutzername lautet. Dies ist die einfachste Methode einen fremden Account zu erhalten. Gegenmaßnahmen:

Server lassen Usereingaben automatisch filtern, wenn *magic_quotes_gpc = on* sowie *magic_quotes_runtime = on* gestellt worden sind in der php.ini . Sollte beides auf off sein, so muss der Programierer unbedingt jede Usereingabe einzeln prüfen. Dafür müsste man in unserem Beispiel folgende Änderung vornehmen:

```
$user_result = mysql_query("SELECT user_id, user_group FROM members
WHERE user_name = '".addslash($user)."' AND user_pass = '".addslash($hash)."' LIMIT
1");
...
```

Durch addslash() wird wie offensichtlich alle Anführungszeichen ' in \' gewandelt und somit für Angriffe unbrauchbar gemacht. Es gibt jedoch noch andere Angriffe über SQL-Injection. Auf diese werde ich hier aber nicht weiter eingehen, sondern verweise hier lieber auf: http://www.unixwiz.net/techtips/sql-injection.html

Solltet ihr noch mehr Informationen zum Knacken von Hackits haben wollen, so schaut einfach bei uns im Forum http://forum.happy-security.de vorbei, dort stehen schon einige Beiträge zu diesem Thema.

Ok, ok... wenn wir jetzt die ganzen Sicherheitslücken ausser Acht lassen und uns denken, es wäre auf die Schnelle erstellt worden, ohne ordentlich getestet zu werden, dann könnte dieses Beispiel auch fast realistisch wirken. Natürlich ist es in der Realität etwas schwerer, aber oft genug können noch solche grob fahrlässigen Fehler gefunden und ausgenutzt werden, wie sich mir in den letzter Wochen immer wieder gezeigt hat :/

Kryptographie

Kryptographie spielt heutzutage beim der regen Benutzung von diversen Internetdiensten eine immer größere Rolle. Kryptographie bedeutet übersetzt soviel wie 'Geheim' und stammt aus dem Griechischen. Mit Kryptologie bezeichnet man die Kunst bzw. die Wissenschaft, Methoden zur Verschlüsselung von Nachrichten zuentwickeln. Die Kryptoanalyse bedeutet hingegen, solch eine Verschlüsselung zu analysieren und entschlüsseln, ohne das Verfahren vorher zu kennen.

Verschlüsselungen gibt es schon über 2500 Jahre und wurde von der Regierung von Sparta als sichere Überbringung von Botschaften genutzt. Selbst wenn ein Feind den Boten überfällt, so konnte er nichts mit der Nachricht anfangen. Gemacht wurde das ganze recht plump mit einem Band und einem dicken Stock (Durchmesser ca. 8-15 cm). Das Band wurde spiralförmig um den Stock gewickelt und dann wurde die Nachricht der Länge des Stockes nach auf das Band geschrieben. Danach das Band wieder abwickeln vom Stock und keiner (der nicht den richtigen Durchmesser des Stockes weiß) kann die Nachricht entschlüsseln. Nagut... es haben dann doch einige etwas rumgetüfftelt und herausgefunden wie es funktioniert. Danach einfach ein paar Stöcke mit unterschiedlichen Durchmesser ausprobiert und das Geheimnis war geknackt. Dies war die Kryptoanalyse und bot danach keinen Schutz mehr gegen Feinde. Es musste also etwas neues her. Aber genug von der Art von Verschlüsselung. Kommen wir zum Internet und den möglichen Arten, die heute noch genutzt werden.

ROT13

Eine der einfachsten Verschlüsselungsarten, die man auch schnell erkennt. Bei ROT13 (ROT steht für ROTation) wird davon ausgegangen, dass nur Texte mit 26 Buchstabenmöglichkeiten verschlüsselt werden. Das heißt: Keine Zahlen, Sonderzeichen oder Umlaute sind erlaubt. Wie der Name schon sagt, rotiert die Verschlüsselung immer um 13 (Buchstaben). Hier ein Beispiel:

```
Klartext:   Hallo Welt. Es ist ein schoenes Wetter.
Geheimtext: Unyyb Jryg. Rf vfg rva fpubrarf Jrggre.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

das ganze läuft auf der Basis unseres Alphabets ab. Es wird einfach jeder einzelne Buchstabe des Klartextes um 13 Buchstaben des Alphabets nach vorne geschoben, um den verschlüsselten Text zu erhalten. Die Groß- und Kleinschreibung wird dabei übernommen. Das 'H' wird also 13 Buchstaben weiter gezählt: IJKLMNOPQRSTU und am Ende sind wir bei dem 'U' angekommen. Wenn man diese Nachricht entschlüsseln will, so macht man es auf den selben Weg rückwärts. Wir merken uns: Wenn eine Nachricht genauso entschlüsselt werden kann, wie sie verschlüsselt

wurde, so sprechen wir von einer **symetrischen Verschlüsselung**. Nun wurde jedoch auch diese Verschlüsselung schnell geknackt und man überlegte sich etwas anderes... man hat einfach anstatt 13 Buchstaben eine Variable Zahl benutzt. So war das entschlüsseln schon sehr schwer (für den Anfang). Das schöne bei solch einer Verschlüsselung wie oben ist jedoch, dass es einige Auffälligkeiten gibt. Da jeder Buchstabe nur einen Partner hat (bei uns das H=U), kann man schnell erkennen, dass einige häufiger vorkommen als andere Buchstaben. Wissenschaftler haben sich viel mit Mathematik und der deutschen Sprache beschäftigt und sind dazu gekommen, wie oft folgende Buchstaben in der deutschen Sprache auftauchen:

Buchstabe	***Häufigkeit***	***Buchstabe***	***Häufigkeit***	***Buchstabe***	***Häufigkeit***
A	6,47	J	0,27	S	6,83
B	1,93	K	1,46	T	6,13
C	2,68	L	3,49	U	4,17
D	4,83	M	2,58	V	0,94
E	17,48	N	9,84	W	1,48
F	1,65	O	2,98	X	0,04
G	3,06	P	0,96	Y	0,08
H	4,23	Q	0,02	Z	1,14
I	7,73	R	7,54		

An Hand unseres Beispieles, kann man sehr gut sehen, wie recht diese Tabelle hat. Der Spitzenreiter 'E' ist auch bei uns der häufigste Buchstabe mit einer Häufigkeit von 7. Ich finde so etwas sehr interessant. Hier noch ein Tipp, für längere Texte: das am meist verwendeste Wort im Deutschen sind „die“ und „und“. Also, wenn in einem verschlüsselten Text ein Wort mit drei Buchstaben vorkommt, so sollte man als erstes von den beiden Wörtern ausgehen (oft liegt man dabei richtig). Wie viele das schon in der Schule bei Hangman mit bekommen haben, sind lange Wörter viel leichter von Schülern zu erraten, als kurze, da die Möglichkeit der Ausnutzung von Häufigkeiten viel eher gegeben ist. Es werden einfach mehr Buchstaben verwendet, was auch heißt, dass mehr von den Häufigsten verwendet werden (a,e,i,n,r,s,t,c+h).

Vigenere-Chiffrierung

Etwas schwerer wird das Ganze nun aber wieder, wenn man eine Art Passwort für das Verschlüsseln nimmt... besser gesagt Schlüssel, da es sich ja auch um eine Ver*schlüssel*ung handelt. Neben dem Schlüssel benötigen wir noch eine Matrix (nee, nicht den Film *grins*). Eine Matrix ist eine Tabelle mit verschiedenen Werten. Wir nehmen für Vigenere eine 26x26-Matrix, da wir ja 26 Buchstaben haben.

0	1	a	b	c	**d**	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	2	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
0	3	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
0	4	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
0	5	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
0	6	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
0	7	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
0	8	**h**	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
0	9	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
1	0	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
1	1	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
1	2	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
1	3	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
1	4	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
1	5	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
1	6	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
1	7	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
1	8	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
1	9	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
2	0	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
2	1	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
2	2	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
2	3	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
2	4	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
2	5	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
2	6	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Dies ist unsere Matrix mit welcher wir jetzt arbeiten... Den Schlüssel den wir nehmen ist 'hcacrew' und der Text, den wir verschlüsseln wollen, kommt hier:

„Dies ist unsere Welt... die Welt der Elektronen.“

Was wir jetzt machen müssen, ist anhand der Tabelle in der ersten Zeile das „D“ (1. Buchstabe des Klartextes) zu suchen und das „H“ (1. Buchstabe des Schlüssels). Dann mit einem Lineal eine Linie bilden, in denen sich beide kreuzen. Bei uns ist es das „k“. jetzt haben wir den ersten Teil der chiffrierten Nachricht. Und so weiter: Jetzt das „i“ und das „c“ suchen und wieder vebinden. Der Buchstabe ist auch ein „k“... dies wird solange gemacht bis wir am Ende des Schlüssels

angekommen sind. Dann einfach wieder am Anfang des Schlüssels (beim „h“) anfangen, bis die ganze Nachricht chiffriert wurd. Das Endergebnis sieht so aus:

Klartext: „Dies ist unsere Welt... die Welt der Elektronen.“
Schlüssel: „hcac ewh acrewh acre... cre hcac ewh acrewhcacr.“
Chiffrier: „Kkec moa upjinl Wgcx... fzi dglv hay Envopypnge.“

Das sieht doch schon richtig Verschlüsselt aus... das ist nicht leicht zu entschlüsseln solange man nicht weiß wie lange der Schlüssel ist. Da hilft nur langes probieren. Auch wenn keine Buchstaben verändert haben, bei denen der Schlüssel „a“ ist, so kann keiner daraus Rückschlüsse ziehen, wenn er nur die Chiffre hat... das ist so cool.

Passwort-Hashing

Auf sogut wie jeder guten Website gibt es heute Login-Bereiche für ihre User, um spezielle Dienste in anspruch nehmen zu können, die man nur bekommt, wenn man registriert ist. Dazu muss jeder einen Username sowie ein Passwort besitzen, mit dem er sich auf der Seite anmeldet. Diese Daten werden fast immer in Datenbanken speichert und beim Einloggen ausgelesen. Da leider oft Konfigurationsfehler bei dem Server auftauchen, können Hacker und Cracker solche Schwachstellen ausnutzen um an den Inhalt der Datenbank zu kommen und schon hätten sie tausende von Passwörtern. Das wäre ziemlich fatal. Aus diesem Grund werden Passwort verschlüsselt in der Datenbank gespeichert... aber nicht mit solch einer „einfach“ zu entschlüsselnden Methode, sondern mit einer Methode, die es nicht ermöglicht das Passwort wieder zurück zu entschlüsseln. Und zwar wird in der Datenbank nur ein Hash-Wert des Passwortes gespeichert. Genauso ein Hash-Wert wird beim Einloggversuch erstellt und mit dem Wert aus der Datenbank verglichen.

-(Registrierung mit dem Passwort „HalloWelt“):
Ergebnis: 476a5533998c2b31c81c2d56a25b83a7
Das Ergebnis wir als Passwort in Datenbank gespeichert.

-(Einloggen mit dem Passwort „Hallowelt“):
Ergebnis: c3f2fd69cbb948aea8fd611d0d4099c2
Das Passwort wird aus der Datenbank genommen und mit dem eben eingegebenen verglichen. Es stimmt nicht überein, also falsches Pass.

-(Einloggen mit dem Passwort „HalloWelt“):
Ergebnis: 476a5533998c2b31c81c2d56a25b83a7
Das Passwort wird aus der Datenbank genommen und mit dem eben eingegebenen verglichen. Der hash stimmt überein mit dem eingegebenen also ist es das selbe Passwort.

Einem Hacker bleibt hierbei nur die Möglichkeit rohe Gewalt anzuwenden. Er muss einfach tausende Mögliche Passwörter mit md5() verschlüsselt (den Hash bilden) und hoffen, dass eins mit dem zu

entschlüsselnden übereinstimmt. Dann wäre es nämlich geknackt. Hier noch mal ein Beispiel: Das Passwort ist „d“ *grins* sehr gut oder? Verschlüsselt wäre es also der Hash 8277e0910d750195b448797616e091ad

a = 0cc175b9c0f1b6a831c399e269772661 = FALSCH
b = 92eb5ffee6ae2fec3ad71c777531578f = FALSCH
c = 4a8a08f09d37b73795649038408b5f33 = FALSCH
d = 8277e0910d750195b448797616e091ad = RICHTIG

a, b, und c waren gehashed nicht 8277e0910d750195b448797616e091ad, aber als aus d der Hash errechnet wurde, war das Ergebnis identisch. Wir haben das Passwort geknackt.

Es gibt noch mehr Verschlüsselungsarten wie md5, aber es würde immer das selbe Prinzip hinter stecken: Mit einer eingebenen Wert (meist Passwort) wird der Hash-Wert gebildet und in Datenbank gespeichert. Danach wird einfach verglichen. Diese Verschlüsselungsarten unterscheiden sich dann in der Anwendbarkeit, also in der Sicherheit und in der Geschwindigkeit. Je sicherer, desto besser, aber auch desto langsamer (grob gesagt).

Sollte noch mehr interesse an Kryptoverfahren und dessen Benutzung bestehen, so schaut einfach mal bei uns auf der Seite vorbei. Wir planen in den nächsten Wochen ein Krypto-Labor zu scripten, in welchem man verschiedene Verfahren testen und diese genauer studieren kann durch nähere Informationen.

Steganographie

Eine andere Möglichkeit, Daten vor dem Zugriff anderer Leute zu schützen und dennoch in Umlauf zu bringen, ist die Steganographie. Bei der Steganographie wird anders als bei der Kryptographie nicht ein Text verschlüsselt sondern viel mehr hinter anderen Dateien ganz einfach versteckt. Das soll heißen, hinter einem einfachen Bild, dass ihr auf der webseite XY als Logo findet, könnte eine geheime Botschaft versteckt sein. Ist für den Unwissenden bei über 1.000.000 Webseiten gänzlich unmöglich eine geheime Botschaft in einem Bild oder sagen wir in einer MP3-Datei zu finden, wenn man nicht genau weiß, wie und wo man anfangen soll. Mittlerweile gibt es sogar wirklich gute Steganographie-Programme im Internet wie zum Beispiel Camouflage, mit welchem man oben genannte Dinge ziemlich professionell machen kann, ohne viel Kenntniss über die Thematik zu besitzen. Aber wenn ihr nur ein Programm benutzen wollt, seid ihr hier falsch ^^. Verstehen heißt die Devise und so gehen wir etwas tiefer. Es geht nämlich auch auf dem einfachen Wege per Hand.

Zum einen ist es oftmals ganz einfach möglich, durchs Öffnen einer Bilddatei mit dem Hexeditor, in der Datei Änderungen vorzunehmen und

kurze, geheime Texte reinzuschreiben. Wichtig ist dabei, dies mit dem Hexeditor zu machen, da sonst die Codierung ins Bildformat später nicht reibungslos klappt und die Datei nicht mehr gelesen werden kann als Grafik. Einfach mal mit dem XVI32-Hexedit rumhantieren, ein Bild einlesen, text irgendwie reinschreiben und wieder speichern.

Eine andere noch ziemlich leichte Möglichkeit des Versteckens von Botschaften in Bildern, folgt nun. Wie in dem Bild rechts, kann man durch das Schreiben von Text in der fast selben Farben wie dem Hintergrund, ihn fast unsichtbar machen. Nur wenn man auf 500% vergrößert, kann man es leicht erahnen, dass dort etwas steht.

Es geht natürlich auch nocht mit einem Programm wie Camouflage. Damit lassen sich sogar ganze Dateien in Bildern verstecken und mit einem Passwort versehen. Das Programm findet ihr auch bei uns im Download-Bereich zum Testen. Aber wie immer ist auch dieser Schutz nicht ultimativ und so gab es schon nach kurzer Zeit ein Gegenmittel, welches den Passwortschutz aushebelt. Zum einen gab es viele Programme, die versuchen die dort verwendete XOR-Verschlüsselung zu brechen und so an die versteckte Datei zu kommen, aber es gibt noch eine viel einfachere Methode ;) Thanks 2 MiB. Angenommen, wir haben ein Bild vor uns liegen, in welchem ein weiteres Bild mit Camouflage passwortgeschützt versteckt wurde. Zu erkennen ist so etwas mit dem Hexeditor, wenn ans Ende der normalen Bilddatei noch ein Haufen Leerzeichen kommen. In Hex-Werten also „20“. Irgendwo zwischen diesem ganzen Empty-Buffer kommt auch dann das verschlüsselte Passwort. In unserem Beispiel ist die jetzt mal der fiktive Wert „45 4D 8F A3“. Man könnte nun also mittels eines Programmes versuchen das ganze zu entschlüsseln ^_^ oder wir terminieren es einfach und ersetzen die XOR-Werte durch ein leeres Passwort. Hier kommt wieder der Wert „20“ zum Einsatz. Angenommen, die Hex-Zeile wäre folgende:

1204: **20 20 20 20 20 20 20 20** 45 4D 8F A3 **20 20 20 20 20 20**

dann könnten wir nun einfach diese Zeile so aussehen lassen:

1204: **20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20**

Nur, wie von Zauberhand, fragt uns das Programm beim Entpacken der geheimen Datei nicht mehr nach einem Passwort. Logisch ;) ist ja auch keins Vorhanden. So einfach können Schutzmechanismen umgangen werden... und nur weil es viele Benutzen, heißt es nicht, dass es auch sicher ist. Benutzt in solch einem Fall am Besten noch zusätzlich den Passwortschutz vom WinRAR. Bei einer Zeichenlänge von 8 Zeichen hat man eine ziemliche Arbeit dies zu entschlüsseln ^_^ Je mehr desto besser natürlich.

Schlußwort:

Ich bin mir sicher, dass es auch dieses mal wieder viele Leute gibt, die etwas gegen dieses Paper auszusetzen haben und meinen, es sei überflüssig bzw. völlig falsch interpretiert.

Dazu kann ich jedoch nur sagen, dass hier einige Information gesammelt und veröffentlicht worden sind, die für Newbies geschrieben wurden. Es soll eigentlich nur als kleine Einleitung zum überdimensionalen Themenbereich des Internets dienen. Wenn einige Leute etwas anderes in dieses Paper hinein interpretieren, dann kann ich leider auch nicht anderes tun, als mit dem Kopf zu schütteln und zu hoffen, dass diese Leute noch einmal über ihre kurz eingefangene Meinung nachdenken und reflexieren.

Allen anderen, die dieses Paper gelesen haben, hoffe ich, dass es etwas geholfen hat, besser in die Welt der Bits und Bytes einzusteigen.

Danksagung:

Vielen Dank an alle Leute, die mich bei dem Schreiben dieses langen Textes unterstützt haben und an mich geglaubt haben. Ganz besonders danke ich an dieser Stelle jedoch folgenden Leuten (in unsortierter Reihenfolge):

HackyD, Vellas, H@kke_peteR, Sputtelkopf, Emac, noother und Sourcerer für die Zusammenarbeit auf unserer Website www.happy-security.de und für das Verbessern dieses Papers. Ebenso danke ich Sputtelkopf für die Unterstützung beim Bau der Website in den ersten Monaten, Lord Kruse für Befreiung vom Unterricht zum Recherchieren im Internet *grins* und Sourcerer für das Durchforsten unserer Website nach neuen Sicherheitslöchern.

Weiter danke ich **Linus Torvalds** für die Entwicklung von Linux, dem **Chaos Computer Club** und der **HE-Crew** (besonders **BlueScreen** & **MCBulba**) die mir erst klar gemacht haben, was richtiges Hacken ist. Special thanks goes to **Marc Ruef** für die coolen Videos und Papers auf seiner Website www.computec.ch. Wo ich gerade bei Tutorials bin: Auch ein großes Lob an **thE_iNviNciblE** für die vielen Tutorials rund ums Hacken (mach mal Pause Alter! mein armer Drucker kann bald nicht mehr ^_^) website: www.the_invincible4ever.de.vu

Referenzen:

Diverse Papers von Marc Ruef von www.computec.ch
„TCP/IP Einsteigerseminar“ by Dirk Larisch [bhv] ISBN-3-8266-7022-1
„Hacken für Dummies“
„Kenne deinen Feind“ by Cyrus Peikari [O'Reilly] ISBN-3-89721-376-1
„Internet Spionage“ by Jack the Hacker [Sybex] ISBN-3-8155-80218
„Secret and Lies“ by Bruce Schneider
„hackerbible 2k“ by Cyberdemon_98
„Hackers Manifeto“ by TheMentor

Erstellt wurde dieses Dokument am 13.04.2004 um 19:06 Uhr und ist am 16.07.2004 um 23:23 Uhr fertig geworden. An diesem Text wurde ungefähr 29 Stunden gearbeitet. UPDATE: nun ist es November 2005 und ich habe keine Ahnung, wie viele Stunden es waren ;) Es bockt auf jeden Fall immer wieder und ich werde auch in Zukunft versuchen hier weiter zu schreiben. Solltet ihr dieses Paper auf eurer Seite als Tutorial anbieten, würde mich das sehr freuen... doch bitte lasst es unverändert und mit Verweis auf unsere Website www.happy-security.de . Danke. Da sich jedoch öfters mal etwas passiert, werde ich versuchen das Dokument so aktuell wie möglich zu halten. Ein Blick in die Tutorial-Sektion von www.happy-security.de lohnt sich also immer.

Weiterführende Links:

Happy-Linkliste	Unser aktuell gehaltene Linkliste
Digital-Library	Unsere Sammlung von Tutorials und mehr.
thE_iNviNciblE_	Gute Tutorials zu allen Themen... schnelle updates
Google	Gute Suchmaschine
HackThisSite.org	Hervorragende Tutorials und Reality-Challenges
www.heise.de	Hier gibt es immer aktuelle Computernews
Wikipedia.org	Wissenslexikon zu gänzlich jedem Begriff

In diesem Sinne noch viel spaß beim Surfen im Netz und schaut ruhig öfters mal bei uns vorbei. Es lässt sich sicher immer etwas neues, interessantes finden.

May the force be with you

Tsutomu Katsura

from www.Happy-Security.de