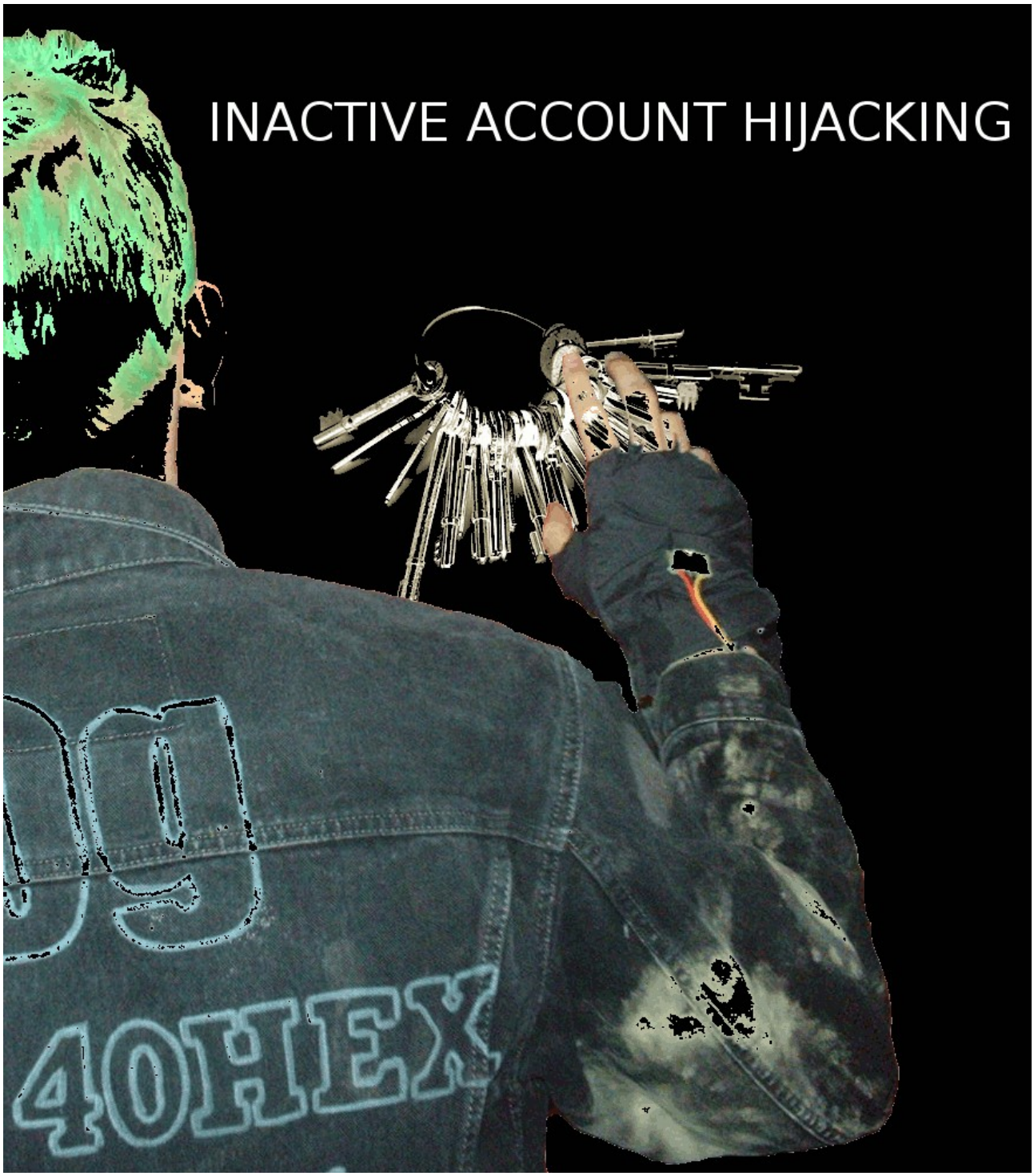
September, 2009



Written by l0om

Version 1.0

Inhaltsverzeichnis

Einführung

Vor nicht langer Zeit habe ich einen Artikel zu einem Thema verfasst, dem ich der Einfachheit halber den Namen „Inactive Account Hijacking“ gegeben habe. Da ich persönlich die ganze Thematik recht interessant finde, habe ich mich dazu entschlossen, ein kleines Tutorial zu diesem Thema zu schreiben. Auch wenn der Name durchweg absurd ist, wie sich später zeigen wird, wollte ich diesen Begriff beibehalten. Hey, die Existenz ist auch absurd, aber beibehalten will ich sie trotzdem ;) Ursprünglich hatte ich geplant, die ganze Sache etwas ernsthafter anzugehen. Letztendlich hatte ich weder Lust noch Zeit.
Beim Inactive Account Hijacking (IAH) handelt es sich um ein einfaches Verfahren, mit welchem sich Zugang zu den Accounts verschiedenster Onlineplattformen erschleichen lässt. Dazu wird ein gelöschter E-Mail Account eines E-Mail Providers erneut registriert, um dann mittels der Passwort-Vergessen-Funktion verschiedener Onlineplattformen Zugriff auf dieselben zu bekommen.
Dabei kommen häufig Fragen auf, wie z.B. Fragen zur Effizienz des Verfahrens oder ob die Ressourcen, auf die ein Angreifer Zugriff bekommen kann, überhaupt schützenswert sind. Die *wenigen* Diskussionen, die ich zu dem Thema IAH finden konnte, habe ich verfolgt und bin dabei mehrfach den gleichen Fragen begegnet. Diese und andere Fragen sollen hier geklärt werden.
Im Weiteren wird natürlich das Verfahren beschrieben, sowie die Ursachen des Angriffs überhaupt und einige theoretische Überlegungen. Damit das Ganze nicht zu trocken wird, sind hier noch einige Programme und deren Ergebnisse enthalten, die die ganze Thematik praktisch vereinfachen.

Selbstverständlich handelt es sich bei diesem Text nicht um eine Aufforderung Straftaten zu begehen, sondern um den Versuch, ein Bewusstsein für das Problem zu schaffen.

Authentifizieren/Autorisierung

Unter Authentifizieren versteht man den Nachweis einer bestimmten Identität. Die Autorisierung erteilt bestimmte Zugriffsrechte auf geschützte Ressourcen. Man unterscheidet für gewöhnlich unter drei Möglichkeiten der Authentifizierung, nämlich Wissen (Passwort, PIN, Sicherheitsfragen), Besitz (Smart Card, Token) und Biometrie (Fingerabdruck, Iris, Retina). Diese Verfahren lassen sich bekanntermaßen miteinander kombinieren, um mehr Sicherheit zu gewährleisten.

Von besonderem Interesse sind in diesem Text Accounts, speziell E-Mail Accounts und Accounts anderer Onlineplattformen, wie Onlineshops oder Foren. Ein Account enthält zumindest Authentifizierungs- und Autorisierungsinformationen, womit sich zum einen ein User Authentifiziert und ihm anhand seiner beglaubigten Identität Autorisierungen für eine Ressource erteilt werden.

Um den Dienst eines E-Mail Providers zu nutzen muss der User seine gewünscht E-Mail Adresse registrieren. Falls die Adresse noch zu haben ist, werden weitere Informationen vom Nutzer verlangt, oftmals handelt es sich dabei um den vollen Namen und die Anschrift. Mir ist kein E-Mail Provider bekannt, bei dem diese Daten tatsächlich geprüft werden. Meist handelt es sich um einen einfachen Abgleich der Daten mit einem Straßen- und Ortsverzeichnis, weshalb die Daten leicht gefälscht werden können. Wenn ein User auf sein Postfach zugreifen möchte, muss er sich mit seiner E-Mail Adresse und seinem Passwort authentifizieren. Dadurch bekommt er Zugriff zu den Ressourcen eines Postfaches, d.h. er kann E-Mails empfangen und (in der Regel) schreiben.

Viele Onlineshops verlangen bei der Registrierung die Anschrift, die E-Mail Adresse und oftmals Angaben für den Zahlungsverkehr. Ein User kann hier Waren kaufen und in manchen Fällen verkaufen. Je nach System sind weitere Informationen zum lesen oder schreiben vorhanden, wie eine Liste der bisherigen und offenen Bestellungen, Nutzerbewertungen, Rechnungs- und Lieferadresse, sowie Informationen zu den Zahlungsmitteln, zum Beispiel Kreditkartennummern oder Angaben zu Bankkonten. Es handelt sich hierbei offensichtlich nicht nur um sensible

Informationen, sondern auch um schützenswerte Funktionen. Es muss sichergestellt werden, dass nicht ein User Bestellungen unter der Identität eines anderen aufgeben kann.

Dagegen bieten Foren oder Online-Games weit weniger kritische Ressourcen, die geschützt werden müssen. Identitätsdiebstahl ist jedoch auch bei den genannten bedingt möglich und es ist offensichtlich, dass auch hier der Schutz der Nutzerkonten von Bedeutung ist.

Schlüsselbund E-Mail Adresse

Verschiedenste Onlineplattformen verlangen für eine erfolgreiche Registrierung die E-Mail Adresse. Diese übernimmt üblicherweise die verschiedensten Funktionen. Vertraut dürfte jedem Nutzer die E-Mail mit dem Bestätigungslink sein. Damit wird sichergestellt, dass der Empfänger der E-Mail, der den Bestätigungslink anklickt, mit der Registrierung des Accounts einverstanden ist. Hier soll also garantiert werden, dass der Besitzer der E-Mail Adresse demjenigen Nutzer zuzuordnen ist, der auch den Versand der Bestätigungsmail mit seiner Registrierung verursacht hat.

Weiter dient die E-Mail Adresse oftmals als Login-Name, der für die Authentifizierung zusätzlich zum Passwort benötigt wird. Das ist besonders häufig bei Onlineshops zu sehen, da dort die Nutzer im Regelfall keinen Nickname verwenden. Bei Foren ist es hingegen eher unüblich, sich mit seiner E-Mail Adresse einzuloggen. Die Verwendung der E-Mail Adresse als Login-Namen hat den Vorteil, dass der Nutzer diese wohl seltener vergisst. Bei Onlineplattformen, wie beim StudiVZ, käme es wohl schnell zu Problemen, wenn der Verwendete User-Name auch Login-Name wäre.

Die E-Mail Adresse fungiert zusätzlich als Kommunikationsschnittstelle zwischen dem Internetdienstbetreiber und dem Nutzer. Der Nutzer empfängt (wohl oder übel) Newsletter Nachrichten, die ihn über Neuigkeiten und wichtige Änderungen informieren sollen. Charakteristisch für Newsletter ist nicht nur, dass sie in den seltensten Fällen gelesen werden, sondern, dass sie besonders zu oder kurz vor Feiertagen verschickt werden.

Eine weitere wichtige Eigenschaft der E-Mail Adresse als Kommunikationsschnittstelle ist die Passwort-Vergessen-Funktion, die sich inzwischen fest etabliert hat. Usus ist, dass ein Nutzer, der sein Passwort vergessen hat, einen Link anklickt, der daraufhin das System veranlasst, eine E-Mail an die hinterlegte E-Mail Adresse zu senden. Innerhalb dieser E-Mail befindet sich ein individueller Link. Wird dieser Link angeklickt, gelangt der Nutzer in der Regel auf eine Maske, auf der er ein neues Passwort für den Account setzen kann.

Wir erinnern uns: bei Bestätigung des Bestätigungslinks, der per E-Mail an eine angegebene E-Mail Adresse versandt wurde, gilt der E-Mail Account Besitzer als identisch mit dem Nutzer, der die Registrierung beim Internetdienst veranlasst hat. Der Besitzer des E-Mail Accounts gilt folglich als autorisiert das Passwort zurückzusetzen und zwar nur, weil er sich bei diesem E-Mail Account authentifizieren kann. Der Prozess der Authentifizierung wird also von der Login-Oberfläche des Internetdienstes auf den Login des E-Mail Providers verlagert.

Inactive Account Hijacking

Im Folgenden wird das allgemeine Angriffsszenario dargestellt und dazu einige Fragen beantwortet. Das ganze Problem soll später noch durch praktisches verdeutlicht werden.

E-Mail Provider

E-Mail Service Provider haben nicht selten viele tausende Nutzer, die alle eine gewisse Anzahl von Speicherplatz für ihr Postfach in Beschlag nehmen und zusätzlich verursacht jedes Postfach Verwaltungsaufwand. Es ist daher nicht verwunderlich, dass E-Mail Provider aller Länder versuchen die E-Mail Postfächer zu löschen, die vom User nicht mehr genutzt werden. Häufig werden E-Mail Accounts nach sechs bis zwölf Monaten Inaktivität gelöscht.

Ich beschränke mich zunächst einmal auf zwei Email Provider, nämlich auf WEB und GMX. Ein Blick in die AGBs von WEB:

[...]
12.4 WEB.DE ist ferner berechtigt,
12.4.1 alle registrierungspflichtigen WEB.DE-Dienste mit sofortiger Wirkung zu kündigen, soweit der Nutzer sein FreeMail-Postfach 6 (sechs) Monate in Folge nicht nutzt.
[...]

Ein Blick in die AGBs von GMX zeigt eine ähnliche Handhabung inaktiver E-Mail Accounts:

[...]
Der Kunde hat für ihn über das Internet eingehende Nachrichten in angemessenen Abständen abzurufen und auf eigenen Rechnern zu speichern. Für kostenlose GMX FreeMail-Tarife gilt darüber hinaus Folgendes: GMX ist berechtigt, die im Account des Kunden gespeicherten Nachrichten und sonstige Dateien nach einem Zeitraum von 6 Monaten der Inaktivität (kein Login über Webbrowser oder E-Mail-Programm) ohne Rückfrage zu löschen. Nach einem Zeitraum von 1 Jahr der Inaktivität ist GMX darüber hinaus berechtigt, die GMX E-Mail-Adressen ("Aliase") des Kunden freizugeben und anderen Kunden zur Verfügung zu stellen.
[...]

Halten wir also fest, dass E-Mail Adressen durchaus ihren Besitzer wechseln können. Dies ist auch nicht selten der Fall. Insbesondere Nutzer, die noch recht neu im Internet sind, legen sich mehrere E-Mail Adressen an. Von diesen bleibt nach einiger Zeit in der Regel nur eine einzige übrig, die anderen geraten in Vergessenheit. Wahrscheinlich hatten die meisten Nutzer bereits mehrere E-Mail Adressen.

Internetdienste

Mit Internetdiensten bezeichnen wir hier Onlineplattformen, die mit dem Browser zu bedienen sind. Viele dieser Internetdienste verlangen, dass sich der User registriert, sich also einen Account, ein Nutzerkonto, anlegt. Dazu zählen z.B. Foren, Onlineshops, Online-Games und soziale Netzwerke.

Interessanterweise ist es bei Internetdiensten, wie Onlineshops, oftmals garnicht möglich, seinen eigenen Account zu löschen oder die Kontakt-E-Mail Adresse zu ändern, mit der man sich registriert hat. Viele Internetdienste haben ein Interesse daran, dass die Accounts ihrer Kunden nie ihre Gültigkeit verlieren. So besteht immer noch die Möglichkeit, dass ein alter Kunde seinen Account reanimiert. Daher werden bei vielen Internetdiensten die Accounts auch nicht gesperrt, auch wenn diese seit geraumer Zeit nicht mehr verwendet wurden. Auch das ständige Versenden der Newsletter hat wohl hier seinen Grund. Der (potentielle) Kunde soll daran erinnert werden, dass er noch beim System X angemeldet ist.

Angriff

Was man sich in Erinnerung holen sollte, ist die Funktion der E-Mail Adresse, wenn sich ein Nutzer unter Angabe dieser E-Mail Adresse bei einem Internetdienst registriert. Ein Nutzer könnte nun in böser Absicht einen freigegebenen Account neu registrieren. Dazu gibt er falsche Namens- und Adressdaten an, die er sich aus jedem Telefonbuch zusammensuchen kann. Wenn dieser Angreifer, wie unterstellt, böse Absichten hat, wird er zur Registrierung einen Proxy verwenden, um möglichst unerkannt zu bleiben.
Nach der Registrierung kann ein Angreifer zunächst einmal bei besonders großen Onlineplattformen nach registrierten Accounts suchen, die diese E-Mail Adresse bei der Registrierung angegeben haben. Dazu reicht es bei der entsprechenden Onlineplattform testweise einen neuen Account anzulegen, wobei als E-Mail Adresse die „gestohlene“ angeben wird. Falls es dabei zu einem Fehler kommt, weil diese E-Mail Adresse bereits verwendet wird, kann der Angreifer sich leicht Zugriff zu diesem Account verschaffen. Alles was er dazu tun muss ist, die Passwort-Vergessen-Funktion für diese E-Mail Adresse zu nutzen.

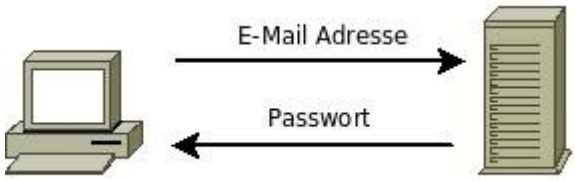


Abbildung 1: Im wesentlichen die übliche Prozedur der Passwort-Vergessen-Funktion (hier Passwort = Zugriff)

Vorläufiges FAQ

F: Sind die Accounts, die ergaunert werden können, überhaupt schützenswert?
A: Es stellt sich tatsächlich die Frage, ob die Accounts, die gestohlen werden, überhaupt weiter schützenswert sind. Wir müssen uns zum klären der Frage einige Dinge ins Gedächtnis rufen:

1. Wo muss man sich registrieren?
2. Mit welchen Daten?
3. Wofür muss ich mich autorisieren?

Registrieren müssen wir uns immer nur dort, wo Daten und Funktionen an eine bestimmte Identität gebunden werden sollen. Je nach Dienst unterscheiden sich nun diese Daten und Funktionen. Ein MySpace Account verlangt so gut wie keine Daten vom Nutzer, bietet unbedeutende Funktionen, erscheint kaum schützenswert. Leider gibt es immer noch Webseiten, die demjenigen, der die Passwort-Vergessen-Funktion ausführt, sein Passwort im Klartext zusendet. Das ist durchaus gefährlich, denn Passwörter werden im Regelfall mehrfach verwendet. Das sind glücklicherweise Ausnahmefälle, die aber auch bei großen Onlineplattformen vorkommen, wie beispielsweise bei MySpace.
Betrachten wir aber einmal Onlineshops. Hier befinden sich sensible Daten und sensible Funktionen, die schützenswert sind, auch wenn der Account des Shops schon lange nicht mehr verwendet wurde. Schützenswerte Daten können hier sein:

- Name
- Anschrift
- Bankverbindung/Kreditkartennummer

- getätigte Bestellungen
- usw.

Welche Funktionen werden durch die Authentifizierung geschützt?
- Bestellungen aufgeben (auf Fremdkosten)
- Lieferanschrift ändern
- Rechnungsanschrift ändern
- Geschäfte mit anderem Account abwickeln

Hier zeigt sich, dass es nicht bloß um völlig nutzlose Nutzerkonten geht.

F: Woher sollte man die E-Mail Adressen nehmen, die wohlmöglich gelöscht wurden?

A: Es gibt verschiedenste Möglichkeiten an E-Mail Adressen zu gelangen. Suchmaschinen liefern mehr Adressen als uns allen Lieb sein kann. Dazu sind nur die richtigen Suchbegriffe nötig. Bekanntermaßen liefert beispielsweise Google auch Ergebnisse, die vom Webseitenbetreiber vielleicht garnicht vorgesehen waren. So finden man neben HTML, PHP und ASP Seiten auch andere Dateiendungen, wie z.B. SQL oder DAT. Bei SQL-Dateien handelt es sich oft um Datenbank-Dumps, mit denen sich Datenbankbestände sichern lassen. Folgender Suchstring wäre also einen Versuch wert:

```
ext:sql intext:"insert into" *.mailprovider.de
```

Ursache des Problems

Wenn ein Nutzer eines Internetdienstes ein Passwort für diese Plattform vergessen hat, kann er sich in der Regel mit Hilfe der Passwort-Vergessen-Funktion wieder Zugriff zu diesem Account verschaffen. Um welche Art der Authentifizierung handelt es sich hierbei eigentlich, wenn man zwischen Wissen, Besitz und Biometrie unterscheidet?
Auch wenn der Nutzer durch den *Besitz* seiner E-Mail Adresse sich gegenüber einer Onlineplattform Authentifizieren kann, handelt es sich hierbei nicht um wirklichen Besitz. Die Authentifizierungsmittel unterscheiden sich dadurch, dass sie völlig getrennte Medien der Identitätsbeglaubigung darstellen. Der Besitz bezeichnet den physikalischen Besitz eines Gegenstands, mit dessen Hilfe man sich Authentifiziert. Angenommen ein Nutzer vergisst also sein Passwort eines Onlinesystems und fordert ein neues Passwort mit der Passwort-Vergessen-Funktion an. Um diese E-Mail lesen zu können muss sich der Nutzer mit seinem Passwort (Wissen) bei seinem E-Mail Account einloggen. Es gibt hier also nur zwei verschiedene Passwörter, die den Nutzer für die selbe Sache authentifizieren und autorisieren.
Die Ursache des Problems ist schnell erkannt. Dazu schauen wir uns eine Vermeidungsstrategie an. Es gibt einige Internetdienste, die der Identität des Besitzers der E-Mail Adresse nicht blindlings trauen. Solche Anbieter stellen dem Nutzer Sicherheitsfragen, wie dies z.B. Ebay tut.

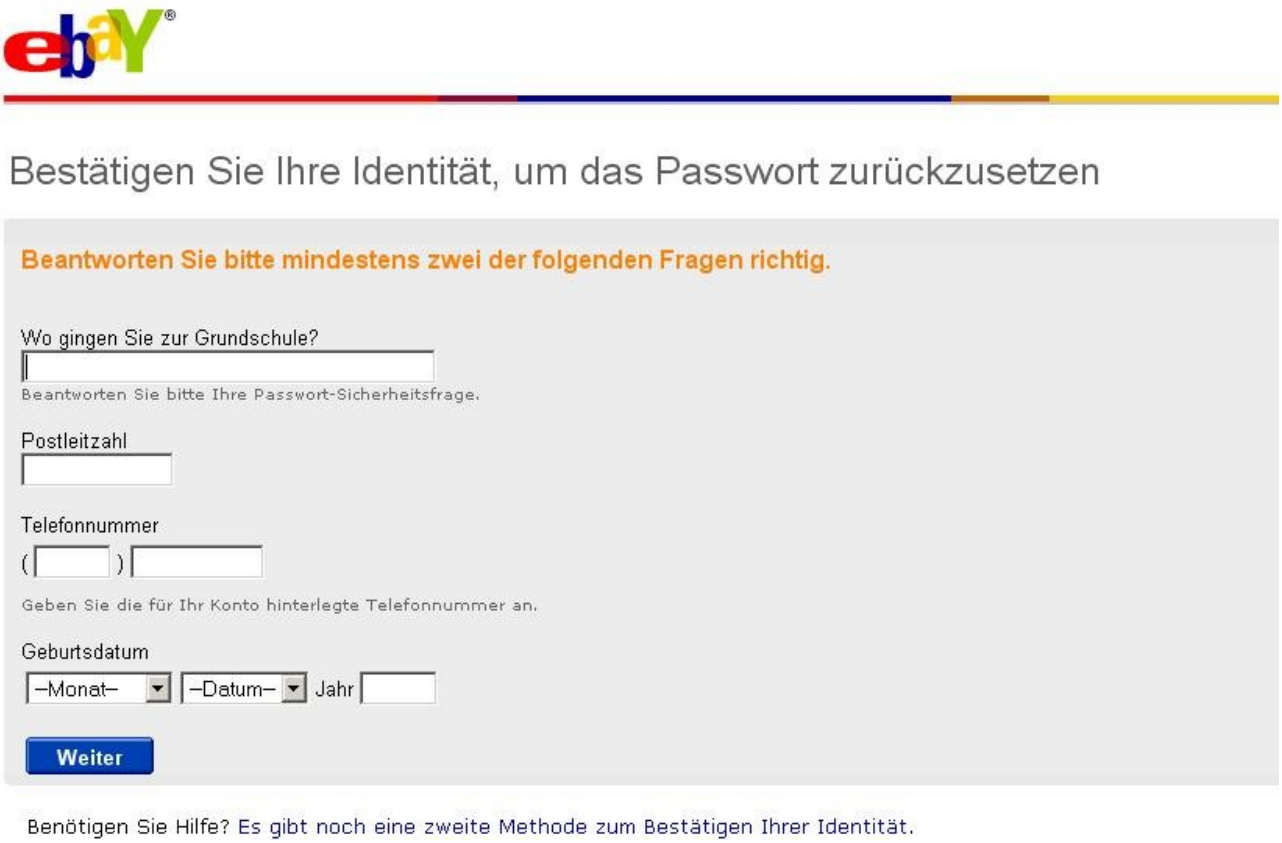


Abbildung 2: Sicherheitsabfragen der Ebay Passwort-Vergessen-Funktion

Zwei dieser Fragen müssen hier also richtig beantwortet werden. Die oberste Frage differiert von User zu User, denn diese muss selbst gewählt werden. Postleitzahl, Telefonnummer und das Geburtsdatum sind jedoch Sicherheitsabfragen für alle Accounts. Das ist eine bessere Praxis als man sie normalerweise vorfindet, doch ist diese Methode sicher und falls nein, weshalb nicht?
Das Internet ist kein sehr schweigsamer Ort. Was man hier hinterlässt, das könnte einen möglicherweise überdauern. Inzwischen gibt es Suchmaschinen, die nach personenbezogenen Daten im Internet suchen. Sie verlangen nur den vollen Namen der Person und schon werden die verschiedensten Quellen durchsucht.

Abbildung 3: Beispiel der Personensuchmaschine 123people.com

Ein ganz wesentliches Problem der oben vorgestellten Sicherheitsabfragen sind die Daten, welche abgefragt werden. Es handelt sich hierbei um Informationen, die sich gut möglich im Internet befinden. Insbesondere in sozialen Netzwerken findet man sehr großzügige Informationen zu verschiedensten Personen, die diese selbst dort hinterlegt haben. Das Geburtsdatum und die Postleitzahl sind durch einen Eintrag in MeinVZ oder StudiVZ schnell ermittelt.
Doch stellen wir uns vor, die Sicherheitsabfrage würde nur sehr sensible Daten erfragen, wie beispielsweise die letzten Stellen der Kontonummer des Besitzers. Falls ein Angreifer bereits Zugriff auf die Daten eines anderen Onlineshops hätte, könnte er selbst solche Informationen aus dem Internet abgreifen. Dieser Fall ist zwar deutlich unwahrscheinlicher, aber dennoch plausibel. Die meisten Internetnutzer haben sich bei mindestens zwei großen Anbietern angemeldet. Mit diesen Überlegungen sind wir der Ursache des Problems sehr nahe gekommen.

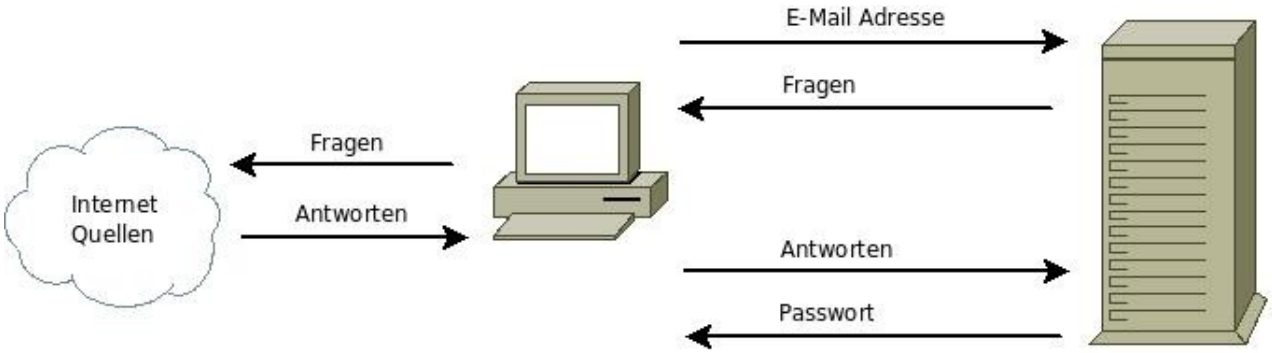


Abbildung 4: Aushebeln der Sicherheitsabfragen durch das Internet

Zur Verifikation des Nutzers muss es mindestens eine Information geben, die nicht durch das Internet beschafft werden kann. Das beinhaltet auch die Nutzung des gestohlenen E-Mail Accounts! Was man häufig antrifft bei Passwort-Vergessen-Funktionen sind Eingabemasken, die nicht die E-Mail Adresse des Accounts verlangen, sondern den Benutzernamen. Falls der Benutzername nicht via Newsletter an den Angreifer und seinen gestohlenen Account gesendet worden sind, ist hier Endstation. Falls es jedoch eine Möglichkeit gibt seinen Benutzernamen beim System abzufragen und zwar mit Angabe der E-Mail Adresse, ist hier natürlich kein Schutz vorhanden.

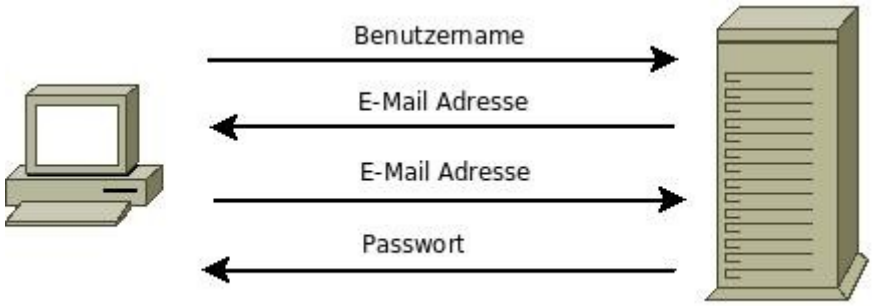


Abbildung 5: Zur Sicherung der Passwort-Vergessen-Funktion muss mindestens eine Information nicht aus dem Internet beziehbar sein!

Es bleibt noch zu klären, ob es sich bei dieser bestimmten Information nicht einfach um ein weiteres Passwort handelt. Dies kann man nicht direkt bejahen, denn im folgenden Beispiel soll

gezeigt werden, dass es sich dabei auch um eine bestimmte Kombination zweier Daten handeln kann, die zwar beide zugänglich sind, aber nicht ohne Zusatzinformationen aufeinander bezogen werden können.

Betrachten wir dazu einige Foren, die im Internet genutzt werden und frei erhältlich sind. Nehmen wir zunächst das populäre **MyBB**. Die Passwort-Vergessen-Funktion verlangt nach einer E-Mail Adresse und sendet dann einen Link an diese Adresse, der das Passwort zurücksetzt. Das genügt nicht den genannten Anforderungen und es kommt noch schlimmer, denn das System gibt zusätzlich Auskünfte darüber, ob ein Account mit der angegebenen E-Mail Adresse beim System existiert oder nicht. Solche Informationen sind bei der Automatisierung von IAH wichtig, doch darauf kommen wir später zu sprechen.

PunBB und **Phorum** arbeiten ebenfalls mit gutgläubigen Passwort-Vergessen-Funktionen. Das **vBulletin** verlangt zumindest die Nutzung eines CAPTCHAs (***C****ompletely* ***A****utomated* ***P****ublic* ***T****uring test to tell* ***C****omputers and* ***H****umans* ***A****part*), wodurch die Automatisierung des IAH unmöglich wird. Als Gewinner sticht das **phpBB** hervor: hier erfragt die Passwort-Vergessen-Funktion nicht bloß die E-Mail Adresse, sondern zusätzlich den dazugehörigen Benutzernamen. Im Forum gibt es keine Rückschlüsse vom Benutzernamen zur E-Mail, allein schon um den Nutzer vor Spammern zu schützen. Ein Angreifer muss sich bei einer solchen Abfrage in der Regel geschlagen geben, aber auch nur, weil der Benutzername nicht mehr zu erfragen ist.

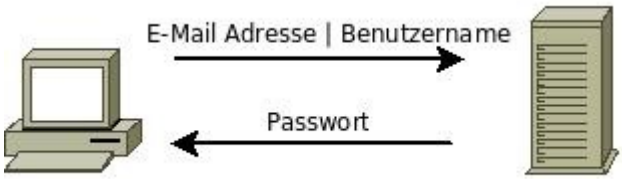


Abbildung 6: Das phpBB verlangt E-Mail und Benutzernamen und ist daher gegen IAH gewappnet

Natürlich ist das nicht das notwendige Ende. Mit einer freundlichen E-Mail an den Betreiber der Onlineplattform ließe sich wohlmöglich doch der Benutzername zur passenden E-Mail Adresse erfragen. Wo wir gerade bei Social Engineering sind: es gibt eine alternative bei Ebay seine Identität bei der Passwort-Vergessen-Funktion zu bestätigen. Dazu benötigt man ein Telefon, wobei man die Telefonnummer selbst angeben kann unter der man angerufen werden möchte. :)

F: *Zwischenfrage: kann überhaupt von einem gestohlenen E-Mail Account die Rede sein und ist der Begriff IAH überhaupt sinnvoll?*
A: Gute Frage. Da der E-Mail Account entweder vom Nutzer selbst oder vom E-Mail Provider gelöscht wurde ist der Account in diesem Zustand nicht mehr vorhanden. Wenn nun jemand den Account erneut registriert, kann wohl kaum vom Diebstahl die Rede sein. Erst von diesem Punkt aus lassen sich andere Accounts hijacken, also „entführen“. Selbst diese Bezeichnung ist an dieser Stelle recht fragwürdig. Es verhält sich also mit der Bezeichnung IAH halbwegs absurd, was mich aber nicht davon abhielt, diesen Begriff zu verwenden. Eigentlich handelt es sich um eine sehr spezielle Art des Identitätsdiebstahls mit allen genannten und noch zu nennenden Konsequenzen.

Newsletter

Explizit möchte ich hier nochmal die Newsletter hervorheben, die für die ganze Thematik wichtig werden können. Wie bereits im ersten Kapitel erwähnt, erhält ein Nutzer eines Internetdienstes

Newsletter des Internetdienstleisters. Diese Newsletter sollen eigentlich informieren und wichtige Systemänderungen mitteilen. Auffällig ist nicht nur, dass kaum jemand Newsletter liest, sondern, dass diese regelmäßig verschickt werden. Wie diese Regel aussieht differiert natürlich stark von Anbieter zu Anbieter, aber spätestens wenn Weihnachten für der Türe steht sendet fast jeder Anbieter einen Newsletter.
Wenn man annimmt, ein Angreifer hat sich bewusst einen gelöschten Account neu registriert, um von dort aus Accounts zu stehlen, dann wird die hier liegende Problematik schnell klar. Zunächst wird ein Angreifer bei großen Onlineplattformen überprüfen, ob dort ein Account mit dieser E-Mail Adresse registriert worden ist. Ist dies nicht der Fall, gibt es immer noch die Möglichkeit, dass der vorherige Besitzer des Accounts sich bei unbekannteren und spezielleren Internetdiensten angemeldet hat. Durch die Newsletter dieser Dienste wird der Dieb quasi darauf hingewiesen, dass hier noch Accounts zu stehlen sind. Abgesehen davon lassen sich oft von hieraus bereits Rückschlüsse zum Vorbesitzer ziehen.

Get it moving!

In diesem Kapitel wollen wir noch wichtige Fragen und Probleme genauer betrachten und das gesamte Konzept mit etwas Leben füllen. Die hier verwendeten Programme sind alle im Anhang dieses Textes zu finden.

Angriffsszenario im Überblick

Fassen wir das kleine Konzept nochmal zusammen, um anschließend die einzelnen Schritte näher zu beleuchten:

1. E-Mail Adressen beschaffen
2. Status der E-Mail Adresse Prüfen
3. Falls E-Mail Adresse freigegeben, diese registrieren
4. Bei Internetdienst überprüfen, ob dort Accounts mit dieser E-Mail Adresse registriert wurden.
5. Falls dies zutrifft, die Passwort-Vergessen-Funktion nutzen.
6. Bei Punkt vier oder eins fortfahren

E-Mail Adressen beschaffen

Hier betrachten wir einige Verfahren, mit denen man an E-Mail Adressen kommen könnte, die wahrscheinlich einmal verwendet wurden.

Suchmaschinen

Es gibt die verschiedensten Suchmaschinen, die allesamt pausenlos das Internet archivieren. Diese Archive können nach bestimmten Begriffen durchsucht werden. Inzwischen bieten Suchmaschinen die Möglichkeit, über einfache Suchbegriffe hinaus seine Suchanfrage zu präzisieren. Dazu gibt es einige Keywords, mit denen man bestimmte Suchergebnisse des Archivs ausschließen kann. Hier ein kurzer Überblick von nützlichen Google-Keywords :

„ext“ oder „filetype“	nur Suchergebnisse innerhalb dieses Dateiformates an.
„intitle“	nur Suchergebnisse von Webseiten mit diesem Titel
„inurl“	die Ergebnis-URL muss diesen String beinhalten
„site“	beschränke die Suche auf diese Domain
„intext“	nur Suchergebnisse, die sich innerhalb der Funddatei befinden
100..4000	alle Ergebnisse mit den Zahlen 100 – 4000 sind gewünscht
-Begriff|Keyword	Negation

Beispiele:

„Zeige mir alle Ergebnisse für @mailprovider.de, in deren Titel der String 'guestbook' und eine Zahl von 1990 bis 2005 enthalten ist!“	intitle:guestbook *@mailprovider.de 1990..2005
„Zeige mir alle Ergebnisse für @mail.de, deren Dateiendung mit 'DAT' endet!“	ext:dat *@mail.de
„Zeige mir alle Ergebnisse für *@mail.org in Dateien mit der Endung 'mbox' und entferne alle Ergebnisse in denen der Teilstring 'sample' vorhanden ist!“	ext:mbox *@mail.org -sample

Da auf diese Art und Weise schon seit langer Zeit Adressen beschafft werden, sind viele Suchanfragen bei verschiedenen Suchmaschinen gesperrt. Zusätzlich arbeiten die meisten Suchmaschinen mit Algorithmen, die auffällige User ausbremsen. Wer also ständig mit dem Muster „*@mail.tld“ nach E-Mail Adressen sucht, wird irgendwann geblockt.

Testweise habe ich dazu ein kleines Python-Skript „emails.py“ geschrieben, dass aus Google-Suchergebnissen E-Mail Adressen extrahiert und ausgibt. Das Skript nimmt optional zum Suchstring noch zwei weitere Parameter entgegen. Diese geben die gewünschten Start- und Endnummer der begehrten Suchergebnisse an. Dadurch können die Suchergebnisse 'weitergeblättert' werden, um mehr Ergebnisse darzustellen.

```
badass@badhost:~$ ./emails.py 'ext:dat *@hotmail.com'
s****s@hotmail.com
R********7@yahoo.com
P*********l@hotmail.com
d*********b@hotmail.com
t*************r@hotmail.com
t*******y@hotmail.com
d**e@handofdoom.com
3*****s@tattoos.com
C*****u@swbell.net
a*****n@earthlink.net
a*****1@prodigy.net
a*************d@hotmail.com
[...]
```

E-Mail Spider

Spider-Programme arbeiten den Suchmaschinen sehr ähnlich. Ein Spider-Programm

1. startet mit einer angegebenen URL
2. durchsucht sie nach E-Mail Adressen
3. speichert gefundene E-Mail Adressen
4. durchsucht die Seite nach URLs
5. folgt jeder URL und beginnt wieder bei (2)

Das ist der gewöhnliche Ablauf, wobei selbstverständlich jede Spider ihre eigenen Eigenschaften bzw. Einstellungsmöglichkeiten hat. Es gibt viele Freeware, Shareware und sogar kostenpflichtige Spider-Programme. Gefunden, aber nicht ausprobiert, habe ich auf die Schnelle „Email Verifier“ und „Email Assault Hun“.

Sicherheitslücken

E-Mail Adressen sind für gewöhnlich Daten, die geschützt werden. Wie wir bereits gesehen haben, werden für Foren oder Onlineshops immer E-Mail Adressen zwecks Registrierung benötigt. Sicherheitslücken in Systemen können nun dafür sorgen, dass die geschützten Daten für Angreifer zugänglich sind. Da ich mal auf eine solche Lücke hingewiesen habe, möchte ich diese ausdrücklich aufführen.

[http://www.securityfocus.com/archive/1/409510]

```
author : l0om innate| @t | gmx.de
WWW.EXCLUDED.ORG        [*seufz*  ;) ]
product: cosmoshop
version: <= 8.10.78
problem:    1. sql injection
            2. cleartext passwords
            3. view any file
maunuf.: www.cosmoshop.de

what is cosmoshop
*****************
cosmoshop is a comercial shop system written as a CGI.

where is the problem
********************

1. sql injection
----------------

the administration login panel suffers from a bad written login function caused
by unfiltered parameters which are put into a sql query. everyone can log in as
admin and can change the pages content. the best/worst of it is: you can
download a mysql dump of the whole shop with the "backup" feature...

other features are:
Article, Columns, Statistics, Supplier, Attitudes, Texts, Design,
Orderprocedure, Mailtexts, Auxiliary-sides, Interfaces, Newletter, Coupons

2. passwords saved in cleartext
-------------------------------

the passwords are stored in cleartext within the database!
[...]
```

In der Datenbank des Shopsystems befanden sich alle Nutzerdaten, wie in dem Advisory explizit genannt, auch das Passwort des Nutzers. Es liegt auf der Hand, dass diese Daten auch die E-Mail Adresse der Nutzer enthielten.

Dies ist nur ein Beispiel, doch die Angriffsmöglichkeiten auf die verschiedensten Websysteme sind enorm. Daher kann hier auch nicht weiter darauf eingegangen werden, aber einige Links sollen dennoch folgen, die dieses Unterkapitel vervollständigen könnten.

CGI Security	Gregory Gilliss:CGI Security Holes, Phrack 49
XSS	GOBBLES: http://www.mail-

	archive.com/bugtraq@securityfocus.com/msg07791.html Erich Kachel: http://www.erich-kachel.de/?p=181
SQL Injection	Stephen J. Friedl: http://unixwiz.net/techtips/sql-injection.html
PHP Security	Jens Ferner: www.tu-chemnitz.de/urz/www/php/rsrc/phpsec.pdf

Status der E-Mail Adresse

Hier soll dargestellt werden, mit welchen Mitteln ein möglicher Angreifer prüfen kann, ob eine E-Mail Adresse zur Zeit registriert ist oder nicht. Hier wird *angenommen*, dass die zu prüfende Adresse einmal registriert war. Falls die Adresse zur Registrierung bereitsteht, rechnet man folglich damit, dass der E-Mail Account zur E-Mail Adresse vom Provider oder vom Nutzer gelöscht wurde.

SMTP

Es hält sich hartnäckig das Gerücht, dass mit SMTP überprüft werden kann, ob bestimmte E-Mail Adressen beim Server bekannt sind oder nicht. Das ist richtig, aber der historische Hinweis auf den VRFY Befehl ist obsolet. Man findet im Internet oftmals Darstellungen wie die folgende:

```
VRFY badass
250 badass@mailprovider.tld
```

Der Returncode 250 zeigt an, dass der Account beim Server gefunden wurde. Es ist allgemein empfohlen, grundsätzlich den Rückgabewert 252 bei VRFY Anfragen zurückzugeben, unabhängig davon, ob die Adresse bekannt ist oder nicht. Man findet äußerst selten SMTP Server, die noch auf VRFY mit 250 antworten. Gleiches gilt für das EXPN Kommando.

Es ist dennoch möglich mit Hilfe von SMTP den E-Mail Status zu prüfen. Dazu loggt sich der Nutzer beim SMTP Server des Mail Providers ein und versendet Mails, die an diese Domain gerichtet sind. Falls der Empfänger nicht bekannt ist, wird normalerweise die Mail mit dem Errorcode 550 verworfen. Anhand dieses Fehlercodes lässt sich also für *diese Domain* feststellen, ob der Account existiert oder nicht. Das Python-Skript „gmx_smtp.py“ automatisiert den erwähnten Vorgang und ist im Anhang A zu finden.

```
badass@badhost:~$ cat emails.db
test@gmx.de
test23232323@gmx.de
asdf232564@gmx.net
badass@badhost:~$
badass@badhost:~$ ./gmx_smtp.py
Account  test23232323@gmx.de  does not exist
Account  asdf232564@gmx.net  does not exist
```

Dies funktioniert allerdings nur bei E-Mail Adressen, die für die Domain des zuständigen Mail Servers vorgesehen ist. So lässt sich also nicht von einem GMX Account aus prüfen, ob ein GMAIL Account registriert ist oder nicht. Sendet man jedoch eine E-Mail an einen unbekannten Empfänger,

wird vom Mail Server der entsprechenden Domain eine Nachricht generiert und an den Absender geschickt. Anhand dieser Nachricht lässt sich also feststellen, ob die Adresse frei ist oder nicht. Bei den Nachrichten handelt es sich um E-Mails, die vom MAILER-DEAMON versendet werden. Dazu wurde ebenfalls ein kleines Programm geschrieben.

```
badass@badhost:~$ ./gmx_mail.py
sent 4 mails
waiting some time before reciving MAILER-DEAMON mails...

looking at 13 mails
received 3 mails from MAILER DEAMON
the following accounts are none existing:
test849238492@gmx.de
kdjaslf5454@hotmail.com
asldkfjad343@web.de
```

HTTP

Angenommen wir möchten einen E-Mail Account bei einem E-Mail Provider registrieren. Dazu müssen wir einige private Daten angeben und anschließend eine Adresse wählen. Natürlich können wir die E-Mail Adresse nur dann wählen, wenn die Adresse noch nicht belegt ist. Also wird das System den Status der Adresse prüfen. Gegenwärtig wird dies oftmals interaktiv mit AJAX realisiert. Es werden also mittels Javascript die benötigten Daten an eine Seite gesendet, die dann je nach Status der E-Mail Adresse antwortet. Man könnte von Hand Adresse für Adresse prüfen. Wirklich interessant sind aber nur Verfahren, die automatisiert werden können.
Um zu erfahren, welche Daten bei der Status Überprüfung genau über die Leitung gehen kann man entweder den Quelltext der Webseite überblicken oder man nutzt einen Sniffer. Anschließend können die benötigten Daten mit einem selbstgeschriebenen oder mit einem vorhandenen HTTP Client an den Server übertragen werden. Als Sniffer bietet sich *Wireshark* an, da sich dieser komfortabel bedienen lässt. Als vorhandenen HTTP Client, der sich leicht für Automatisierungen nutzen lässt, empfehle ich CURL. Das Bash-Skript „hippolyte.sh“ überprüft unter anderem den Status verschiedener E-Mail Accounts über HTTP.

Wireshark:
[http://www.wireshark.org/]

CURL:
[http://curl.haxx.se/]

E-Mail Adresse registrieren

Eine E-Mai Adresse registrieren kann man eigentlich nur unter Verwendung eines CAPTCHAs. Dabei handelt es sich um dynamisch generierte Bilddateien, die Buchstaben enthalten. Diese Buchstaben sind in der Regel verzerrt, sind zwar für Menschen lesbar, aber Computer sind meist damit überfordert. Viele CAPTCHAs sind zur Zeit nicht zu analysieren, aber es gibt immer wieder Ausnahmen. Um einfachere CAPTCHAs automatisch zu analysieren empfehle ich die Verwendung des GAMERA-Frameworks. Dabei handelt es sich um einen großen Funktionsumfang, um den die Skriptsprache Python erweitert werden kann. Anschließend können Zeichen heraussegmentiert und klassifiziert werden. Es handelt sich hierbei jedoch um ein sehr umfangreiches Themengebiet, auf das ich daher hier nicht weiter eingehen kann.

Recht aktuelles Beispiel:
[http://www.heise.de/newsticker/Spammer-hebeln-Googles-Captchas-aus--/meldung/104854]

Das Gamera-Framework und Tutorials:
[http://gamera.informatik.hsnr.de]

Was noch der Vollständigkeit halber zu erwähnen ist, ist die Tatsache, dass die Persönlichen Daten, die für die Registrierung verlangt werden, oft nicht Wahrheitsgetreu angegeben werden. Die verwendeten Daten stammen oftmals aus Onlinetelefonbüchern oder der Phantasie. Auf Seiten der Internetdienste stellt die Überprüfung dieser Angaben auch ein Problem dar. Bei E-Mail Providern erschöpft sich die Kontrolle in der Überprüfung der Existenz der Adresse oder der angegebenen Postleitzahl.

Account beim Internetdienst prüfen

Falls der überprüfte E-Mail Account registriert werden kann, wird ein Angreifer prüfen, ob dieser E-Mail Account bei der Registrierung bei verschiedenen Internetdiensten angegeben wurde. Wie bereits erwähnt handelt es sich bei diesen Internetdiensten um die verschiedensten Angebote, wie Onlineshops, Foren oder Onlinegames.

Neue Registrierung

Ob eine E-Mail Adresse bei der Registrierung eines Accounts bei einem Internetdienst verwendet wurde, lässt sich leicht feststellen, indem ein neuer Account mit gerade dieser Adresse angelegt wird. Falls die Adresse schon im System hinterlegt wurde, wird die Registrierung mit dem entsprechenden Hinweis fehlschlagen. Somit lässt sich grundsätzlich jede Adresse an einem System prüfen, aber hier verhält es sich oft schwierig. Da nämlich verhindert werden soll, dass automatisiert Accounts angelegt werden, wird die Registrierung im Regelfall nur unter Verwendung eines CAPTCHAs gelingen.

Passwort-Vergessen-Funktion

Manche Systeme geizen nicht gerade mit Informationen. So gibt es Passwort-Vergessen-Funktionen, die bei der Eingabe einer nicht vorhandenen E-Mail Adresse die Information ausgeben, dass kein Account mit dieser Adresse gefunden werden konnte. Falls ein System so großzügig Informationen preis gibt, können diese Informationen in der Regel auch automatisiert abgefragt werden. Nur in seltenen Fällen wird zur Verwendung der Passwort-Vergessen-Funktion eine CAPTCHA Abfrage eingebaut.

Abbildung 7: Informative Passwort-Vergessen-Funktion vom StudiVZ

Zugriff auf Account anfordern

Zu diesem Punkt lässt sich nicht viel sagen. Die meisten Passwort-Vergessen-Funktionen lassen sich ohne CAPTCHA bedienen und verlangen nur die E-Mail Adresse des Accounts. Hier ist jedoch der entscheidende Moment zu finden, bei dem potentielle Angreifer ausgebremst werden können. Verlangt die Passwort-Vergessen-Funktion nämlich Daten, die ein Angreifer nicht kennt, kann er diese Funktion logischerweise nicht nutzen.

Wegwerfadressen

Auf ein Thema möchte ich in diesem Kapitel noch explizit zu sprechen kommen, nämlich die Wergwerf-E-Mail Adressen. Im Internet gibt es viele Anbieter, die solche E-Mail Accounts zur Verfügung stellen. Dabei handelt es sich um E-Mail Accounts, die mit eintreffen einer Mail, die für diese Domain vorgesehen ist, erzeugt werden. Wenn also die Mail „john-rockt“ bei dem Wegwerf-E-Mail Provider „fakemailaccount.org“ eintrifft, wird temporär ein Account für die Adresse „john-rockt@fakemailaccount.org“ eröffnet. Dieser Account kann von jeder Person eingesehen werden. Der Account bietet jedoch keine Möglichkeit, von hier aus E-Mails zu versenden. Sinn und Zweck solcher Accounts ist es, dass man beispielsweise Newsletter empfangen und lesen kann, ohne seine echte E-Mail Adresse preis zu geben, die möglicherweise so in Umlauf geraten könnte.
Ein Blick in die FAQs oder AGBs solcher Dienstleister macht schnell deutlich, dass diese Accounts keinerlei Sicherheit garantieren, sondern dass alle Accounts für alle einsehbar sind. Jede Information auf solchem Account ist Allgemeingut. Sinnvollerweise sollte man erstens keine wichtigen Informationen auf solch einer E-Mail Adresse erwarten, noch sollte man einem solchen E-Mail Account eine ernsthafte Funktion übergeben.

Abbildung 8: Wegwerf E-Mail Adressen Provider

Es bleibt jedoch die Frage offen, ob sich wirklich jeder Nutzer darüber im Klaren ist, dass diese Accounts für jeden einsehbar sind. Dazu wurde ein kleiner Test durchgeführt. Es wurden zwei Wegwerf Adressen Anbieter mit je 200 Vornamen gefüttert. Die Ergebnisse wurden gespeichert und später ausgewertet.

Provider	**Getestete Accounts**	**Leere oder unbenutzte Accounts**
A	200	94
B	200	101

Von 400 überprüften Accounts waren also 195 ungenutzt. Es blieben also die Daten von 205 Accounts, die zu überprüfen waren. Die Ergebnisse wurden mit dem Schlüsselwort „Newsletter“ (größeninvariant) durchsucht, um Rückschlüsse auf Internetdienste machen zu können, bei denen sich ein Nutzer möglicherweise mit Verwendung eines solchen Accounts angemeldet hat.

Art des Internetdienstes	**Anzahl der möglichen Account-Hijacks**
Soziale Netzwerke	4
Onlinegame	1
Dating	1
Sonstiges	1

Zumindest sind hier keine Shop Accounts angelegt worden. Auf 207 Accounts mit Inhalt kamen also 7 mögliche Diebstähle.

Internet-Service-Provider E-Mail Adressen

Worauf ich in diesem Zusammenhang ebenfalls zu sprechen kommen möchte, sind E-Mail Adressen, die man als Kunde eines ISP erhält. Viele große ISP weisen einem Neukunden eine bestimmte E-Mai Adresse zu. Wenn sich z.B. Max Mustermann bei einem ISP seiner Wahl anmeldet, kann er eine E-Mail Adresse wie „max.mustermann@your-isp.com“ erhalten.
Wirklich interessant wird die ganze Thematik nun, wenn der ISP auch gleichzeitig einen Freemail-Dienst anbietet und dieser die gleiche Domain besitzt. Wenn nämlich Max Mustermann seinen ISP wechselt, wird auch seine E-Mail Adresse aufgelöst. Es ist wohl jedem bekannt, dass der Wechsel von einem ISP zum anderen sehr häufig aufgrund des Preiskampfs geschieht. Falls hier die Möglichkeit besteht den Account „max.mustermann@your-isp.com“ neu zu registrieren, kann die Verkettung von ISP mit Freemail-Dienst üble Folgen haben.

IAH – Why nobody gives a fuck

Es stellt sich die Frage, weshalb IAH niemanden (außer mir) interessiert :) Dieser Frage soll hier mit einer kleinen Statistik nachgegangen werden um quantitativ zu zeigen, von welcher Gefahr hier eigentlich die Rede ist.

Voraussetzungen

Untersucht wurden E-Mail Accounts zweier E-Mail Provider, nämlich GMX und WEB. Die E-Mail Accounts sind alle *mindestens* sechs Jahre alt. Nähere Angaben zum Erstellungsdatum der E-Mail Adressen können hier nicht gemacht werden. Es ist möglich, dass die eine oder andere Adresse doppelt vorkommt, doch diese Anteile dürften marginal sein.
Überprüft wurde die *theoretische Möglichkeit*, von einer besagten E-Mail Adresse aus einen Account bei den Internetdiensten Amazon, Dawanda, StudiVZ und MySpace zu stehlen. Diese Internetdienste haben selbstverständlich keinen Sicherheitsschutz gegen das vorgestellte Verfahren, sonst wären die Ergebnisse sinnlos.

Ergebnisse

Provider	**Überprüft**	**Freigegeben**	**Relative Häufigkeit**
GMX	2300	269	0,12
WEB	3136	141	0,31

Im Weiteren werden die freigegebenen Accounts als Dreh- und Angelpunkt für theoretische Account-Diebstähle betrachtet.

Internetdienst	**E-Mail Provider**	**Mögl. Diebstähle**	**Relative Häufigkeit**
Amazon	GMX	84	0,31
Amazon	WEB	29	0,21
MySpace	GMX	1	0,004
MySpace	WEB	1	0,007
Dawanda	GMX	0	0,0
Dawanda	WEB	0	0,0
StudiVZ	GMX	0	0,0
StudiVZ	WEB	0	0,0

Bei den betrachteten statistischen Ereignissen handelt es sich um unabhängige Ereignisse. Die Wahrscheinlichkeit, dass ein E-Mail Account frei ist, hat keinen Einfluss darauf, ob ein freier Account auch für die Registrierung bei einem Internetdienst verwendet wurde. Wenn wir also die Gesamtwahrscheinlichkeiten für das Eintreten beider Ereignisse betrachten wollen, können wir die relativen Häufigkeiten miteinander multiplizieren.
Wie wahrscheinlich ist es also nun, dass eine E-Mail Adresse freigegeben wurde und dass dieser Account für die Registrierung bei einem Internetdienst verwendet wurde?

$$P(GMX\ frei) * P(AmazonRegistiert) = 0{,}0372$$
$$P(WEB\ frei) * P(AmazonRegistiert) = 0{,}008$$

Bei diesen Werten kann man verstehen, dass man lieber nichts tut, als dass man zusätzliche Arbeit auf sich nimmt. Für Angreifer hat dies nur den Vorteil, dass das beschriebene Verfahren wohl auf lange Sicht funktionieren wird.
Es bleibt jedoch noch die Frage offen, welche Rückschlüsse sich von den hier vorgestellten Ergebnissen auf die unbetrachteten E-Mail Accounts machen lassen. Dazu approximiert man das Konfidenzintervall, wobei für die hier betrachteten Ereignisse als Verteilungsmodell die Binomialverteilung gewählt wurde. Die Binomialverteilung lag auf der Hand, da wir binäre Ereignisse betrachten, die wie ein Münzwurf entweder wahr oder falsch sind. Dafür ergibt sich:

$$\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

$\hat{p}$ beinhaltet die realtive Häufigkeit, die sich aus den getesteten Accounts (n) und der Fehlerrate ergibt. Mit α von %5 ergibt sich für $z_{1-\frac{\alpha}{2}}$ der Wert 1,96. Durch α von 5% errechnen wir den die realtive Häufigkeit für 95% aller Fälle. 5% der noch zu untersuchenden Fälle werden jedoch abweichen – *statistisch*! Und statistisch auch nur, wenn die genannten Tests der weiteren 95% der E-Mail Accounts unter den gleichen Bedingungen stattfinden. Dabei muss man hier nochmals *genau* Rücksicht auf die anfangs genannten Bedingungen nehmen.

Wahrscheinlichkeit für gelöschten GMX Account unter Berücksichtigung des Konfidenzintervalls ist:

$0{,}12 \pm 0{,}0133$

Wahrscheinlichkeit für gelöschten WEB Account unter Berücksichtigung des Konfidenzintervalls ist:

$0{,}31 \pm 0{,}0162$

Wahrscheinlichkeit auf die Möglichkeit, einen Amazon Account durch einen freien GMX Account zu stehlen:

$0{,}31 \pm 0{,}1$

Anhang A

Es handelt sich bei den folgenden Python Programmen um Quick-And-Dirty Lösungen.

emails.py

```
#!/usr/bin/python

# emails.py  -  l0om
#
# how to get email addresses from google?
# like this.
#
# this is a example python code and it was written as a part of a paper
# about inactive account hijacking.
#

import sys
import re
import httplib
import urllib

def main(argv):
  start=0
  end=10

  if(len(argv) == 3):
    start = int(argv[1])
    end = int(argv[2])

  tmp = []
  while(start < end):
    h1 = httplib.HTTPConnection("www.google.de")
    request = urllib.quote(argv[0]);
    start_url = urllib.quote(str(start))
    h1.request("GET", "/search?q="+request+"&start="+start_url)
    r1 = h1.getresponse()

    if(r1.status != 200):
      print "HTTP Error"
      print str(r1.status) + r1.reason
      sys.exit(1)

    data = r1.read()
    data=data.replace(' ','')
    data=data.replace('<em>','')
    data=data.replace('</em>','')
    #print data
    res = re.findall('[a-zA-z0-9._+-]+@[a-zA-Z0-9.-]+\.(?:de|com|net|org|biz)', data)
    if(len(res) == 0):
      print "NO results"
    else:
      if(len(tmp) > 0):
       for i in res:
         if(i in tmp):
           tmp.remove(i)
       if(len(tmp) == 0):
         print "guess i received all results. stopping."
         sys.exit(1)
      for i in res:
       print i
    tmp = res
    h1.close()
    start = start + 10

if __name__ == "__main__":
    main(sys.argv[1:])
```

gmx_smtp.py

```
#!/usr/bin/python
```

```
import smtplib
import sys

def main():
  username="the_account@gmx.de"
  userpassword="asdf44asdf"
  email_db_file="emails.db"

  f = open(email_db_file, "r")
  filestuff = f.read()
  addresses = filestuff.split("\n")
  if(len(addresses) == 0):
    print "no emails"
    sys.exit(1)

  send = smtplib.SMTP("mail.gmx.net")
  #send.set_debuglevel(1)
  send.starttls()
  send.login(username, userpassword)

  for mailto in addresses:
    msg = "To: "+mailto+"\n"
    msg = msg + "From: "+username+"\n"
    msg = msg + "Subject: asdf\n\n"
    msg = msg + "asdf" + "\n"
    try:
      send.sendmail(username, mailto, msg)
      #print "have sent mail to " + mailto
    except smtplib.SMTPRecipientsRefused:
      print "Account ", mailto , " does not exist"
  send.quit()

if __name__ == "__main__":
    main()
```

gmx_mail.py

```
#!/usr/bin/python

import smtplib
import poplib
import sys
import re
import time

def main():
  username="the_account@gmx.de"
  userpassword="asdf44asdf"
  email_db_file="emails.db"
  sleep_seconds = 2

  f = open(email_db_file, "r")
  filestuff = f.read()
  addresses = filestuff.split("\n")
  if(len(addresses) == 0):
    print "no emails"
    sys.exit(1)

  sent = send_mails("mail.gmx.net", username, userpassword, username, "asdf", addresses)
  print "sent "+ str(sent) + " mails"
  print "waiting some time before reciving MAILER-DEAMON mails..."
  time.sleep(sleep_seconds)
  non_existing = get_mails("pop.gmx.net",995, username, userpassword)
  print "received " + str(len(non_existing)) + " mails from MAILER DEAMON"

  print "the following accounts are none existing:"
  for address in non_existing:
    print address


def send_mails(server, user, password, mailfrom, mailcontent,addresses):
  i=0
  send = smtplib.SMTP("mail.gmx.net")
  #send.set_debuglevel(1)
  send.starttls()
```

```
  send.login(user, password)

  for mailto in addresses:
    msg = "To: "+mailto+"\n"
    msg = msg + "From: "+mailfrom+"\n"
    msg = msg + "Subject: asdf\n\n"
    msg = msg + mailcontent + "\n"
    try:
      send.sendmail(mailfrom, mailto, msg)
      print "have sent mail to " + mailto
    except smtplib.SMTPRecipientsRefused:
      print "Mailaddress " + mailto + " is free"
    i = i + 1
  send.quit()
  return i

def get_mails(servername, port, user, password):
  poppen = poplib.POP3_SSL(servername,port)
  poppen.getwelcome()
  #poppen.set_debuglevel(1)
  poppen.user(user)
  poppen.pass_(password)
  l = poppen.list()
  mails=len(l[1])
  print "looking at "+str(mails)+" mails"

  addresses = []
  for i in range(mails):
    i = i + 1
    mail = poppen.retr(i)
    mail_str = str(mail[1])
    if(mail_str.rfind('MAILER-DAEMON@mail.gmx') != -1):
      print "Mail "+str(i)+" ist interessant"
      m = re.search('<[a-zA-Z0-9_\-\.@]+>:', mail_str)
      if(len(m.group(0)) == 0):
        continue
      address = m.group(0)[1:-2]
      addresses.append(address)
  poppen.quit()
  return addresses

if __name__ == "__main__":
    main()
```

hippolyte.sh

If you are looking for this script check out the FREE WORLD.

http://packetstormsecurity.org/Crackers/hippolyte.txt