

# Sikkerhet på Web

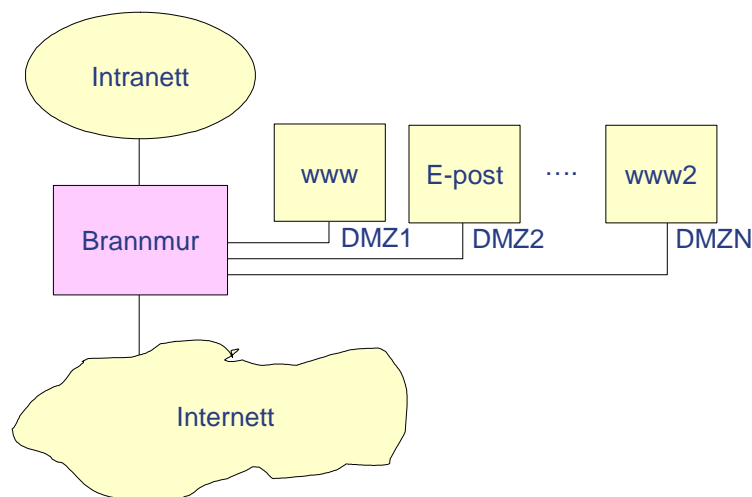
Kåre Presttun  
Software Innovation ASA  
[Kare@Presttun.org](mailto:Kare@Presttun.org)  
<http://www.presttun.org/kare/>

Denne artikkelen vil gjennomgå forskjellige aspekter ved sikkerhet på web og blant annet drøfte brannmur arkitektur og konfigurasjon, sikkerhet på selve web serveren, web autentisering og SSL. Det blir også gitt en oversikt over verktøy som kan brukes til å sjekke sikkerheten. Til slutt blir det en gjennomgang av overvåkningsmetoder som Intrusion Detection Systemer (IDS) og Intrusion Prevention Systemer (IPS).

Artikkelen går ikke inn på området sikkerhet i web applikasjoner som er et stort felt og like viktig for sikkerheten som infrastrukturen.

## 1 Brannmurer

Brannmurer brukes til å beskytte bedriftens interne nett mot de farene som lurer på Internett. Ofte ønsker man å eksponere tjenester som e-post og web mot



Figur 1: Brannmur med oppdelt DMZ

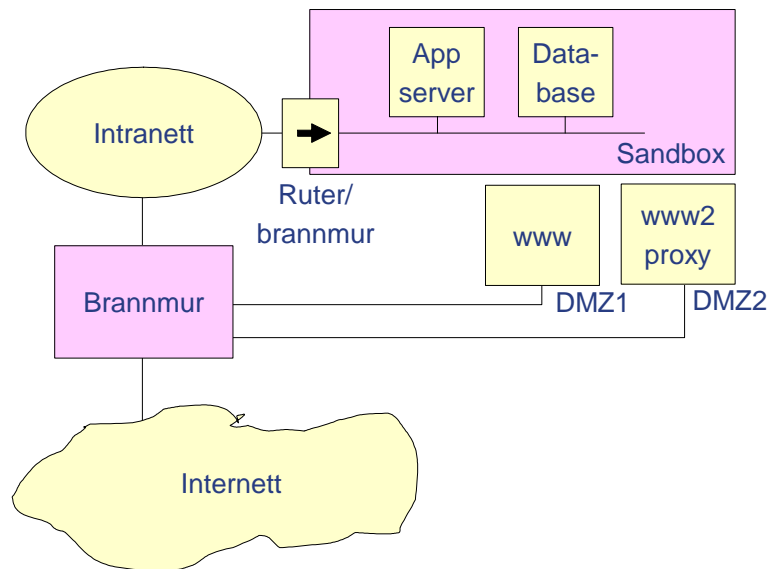
omverdenen. Normalt gjøres det ved å opprette et eget nettverk på brannmuren hvor en setter opp de tjenerne som skal tilby disse tjenestene. Et slikt nettverk kalles demilitarisert sone (DMZ). På et slikt DMZ nettverk kan det være mange tjenere som brukes for mange forskjellige formål. Noen skal kanskje bare eksponeres for en gruppe som f.eks. kunder. Hvis en da setter alle tjenerne på ett og samme DMZ nettverk så vil et innbrudd i en av dem enkelt kunne forplante

seg til de andre tjenerne. På DMZ nettverket er de ikke beskyttet mot hverandre. Da er det en fordel å dele opp DMZ nettverket slik som vist i Figur 1. På den måten vil serverne på DMZ nettverket være beskyttet mot hverandre. Dermed vil heller ikke et innbrudd så lett kunne forplante seg.

Det er også viktig at en ikke åpner for flere tjenester i brannmuren enn det som er nødvendig for at det skal virke. Når det gjelder web betyr det stort sett TCP (transmission control protocol) port 80 for vanlig web trafikk og TCP port 443 for kryptert trafikk (https). Når en konfigurerer brannmurer er det viktig å huske på en annen av IP protokollene nemlig ICMP (internet control message protocol). Jeg anbefaler at en også åpner for ICMP unreachable - fragment needed (type3 code4)

og ICMP source-quench (type4 code0). Dokumentet "TCP Problems with Path MTU Discovery" diskuterer en del problemstillinger rundt fragmentering [1].

I mange tilfeller er det ikke ønskelig eller mulig å legge hele web løsningen ut på DMZ. Da kan man legge web fronten eller en proxy på DMZ nettet og ha



Figur 2: Interne servere isolert i sandbox

baksystemene som f.eks. databaser stående på det interne nettet. Faren med å gjøre dette er at hvis web fronten blir kompromittert så har angriperen adgang til den interne serveren og via den potensielt til hele det interne nettet i organisasjonen. Nok en gang er det viktig å dele opp systemene slik at en begrenser mulig skade. En måte å gjøre dette på er vist i Figur 2. Her er brannmuren satt opp til å kun tillate proxy å snakke med intern applikasjonsserver. Denne står på et nett som kalles

sandbox som er isolert fra resten av det interne nettet med en ruter eller en brannmur. Denne tillater inngående trafikk fra proxy samt fra intranettet for administrasjon. Den tillater ikke trafikk initiert fra sandbox-nettet å nå Intranettet. Dermed kan ikke servere i sandbox-nettet brukes som mellomstasjon for å angripe Intranettet.

Det grunnleggende prinsippet både med multiple DMZ soner og med sandbox-nettet er å begrense mulig skade så mye som mulig og samtidig tilby nødvendig funksjonalitet.

## 2 Autentisering

Ofte er det ønskelig at brukere skal logge seg på tjenesten og autentisere seg. Det finnes mange måter å gjøre dette på i websammenheng. De vanligste metodene er:

1. http med vanlig brukernavn og passord
2. https med vanlig brukernavn og passord
3. Samme med engangspassord (passordkalkulator)
4. https med klientsertifikat (toveis SSL/TLS)

Den første er den enkleste og mest usikre. I tillegg til å bruke http autentisering er det også vanlig å bruke skjema-login. Da vil en få opp brukernavn- og passordfelt på websiden istedenfor at det popper opp en påloggingsboks. I begge tilfellene sendes brukernavn og passord over nettet i klartekst og kan derved snappes opp av uvedkommende. Derfor er det ganske vanlig å tilby pålogging over kryptert forbindelse (https).

Pålogging både via klartekst og kryptert forbindelse kan også gjøres med engangspassord. Dette er vanlig i dag i nettbanker. Fordelen med å bruke

engangspassord er at om passordet blir snappet opp av uvedkommende så kan det ikke brukes fordi brukeren må ha et nytt neste gang han logger på. Det betyr naturligvis at brukeren må ha noe verktøy for å lage disse passordene enten programvare eller maskinvare. Det finnes mange slags systemer for dette. En gratis løsning i programvare heter OPIE [2] (One Time Passwords in Everything). Det finnes også mange kommersielle løsninger som f.eks. Safeword, SecureID og Digipass.

Den sikreste løsningen er å utstyre brukerne med nøkler og sertifikater slik at de kan autentisere seg direkte ved oppkobling via https dvs. SSL (Secure Socket Layer) eller TLS (Transport Layer Security [3]). Dette begynner nå å komme i bruk. Eksempler er online spilltjenesten til Norsk Tipping og BankID som bankene er i ferd med å innføre.

En annen side ved å ta i bruk toveis autentisering via SSL eller TLS er at det hindrer mann i midten (MitM) angrep. Ved vanlig kryptert oppkobling uten klientnøkler er slike angrep mulig og programvare for å automatisere slike angrep er tilgjengelig [4].

Det er også ganske vanlig å lage en autentiseringstjener løsning for å håndtere autentiseringen. Fordelen med å gjøre det istedenfor å legge autentiseringsløsningen på web tjeneren selv er at den infrastrukturen en autentiseringstjener er kan brukes til andre formål som ansattes VPN aksess eller tilgang til interne applikasjoner. Slike autentiseringstjenere er normalt basert enten på LDAP (Lightweight Directory Access Protocol [5]) eller Radius (Remote Authentication Dial In User Service [6]). En slik tjener må også plasseres i sandbox.

Når brukeren først er autentisert brukes ofte en sessionID mekanisme for å vedlikeholde sesjonen siden http er tilstandsløs. Her har utviklere gått i mange feller. En slik ID må være random og tilhøre en stor mengde. David Endler diskuterer dette inngående [7].

### 3 Kryptering (SSL/TLS)

SSL er en proprietær krypteringsløsning utviklet av Netscape. Protokollen er implementert i de fleste nettlesere og svært utbredt. TLS er en standardisert utgave. Den er ikke direkte kompatibel med SSL men har en fall-back modus til SSLv3. Begge tillater toveis autentisering med klient nøkler og sertifikat.

Et problem med klientnøkkel er beskyttelsen av nøkkelen. Hvis en brukers private nøkkel kommer på avveie kan noen gi seg ut for å være vedkommende. Hvis nøkkelen ligger beskyttet i programvare er det flere muligheter for at trojanere kan plukke den opp. Vanligvis ligger nøkkelen kryptert under en nøkkel som er avledet av brukerens passord. Den private nøkkelen vil også være i klartekst i maskinens hukommelse et lite øyeblikk i det den blir brukt. Trojanere som utnytter den siste muligheten er ikke kjent utover en "proof of concept" implementering [8]. Alle disse mulighetene for angrep på den private nøkkelen er en pådriver for å bruke smartkort. Ved å bruke smartkort i PKI sammenheng vil brukerens private nøkkel aldri forlate smartkortet. All prosessering med nøkkelen vil også foregå i smartkortet. Trojanere som plukker opp tastetrykk vil fremdeles finne PIN koden men det hjelper ikke uten fysisk tilgang på kortet. Sikkerheten blir følgelig høyere. Det finnes også løsninger der all prosessering og nøkler finnes i et senter på nettet. Slike løsninger kan også gi

god sikkerhet men et slikt senter blir en kritisk komponent for tilgjengelighet av tjenesten.

På tjenersiden er både sikkerhet og ytelse noe en må ta hensyn til. Kryptering er ganske intensivt på CPU-forbruket. Hvis tjenerens private nøkkel ligger beskyttet i programvare er den sårbar for forskjellige typer angrep på samme måten som på klientsiden. Smartkort er ikke noen løsning på tjenersiden pga. ytelse, det må kraftigere lut til. Løsningen ligger i såkalte akseleratorer. Til webbruk finnes mange typer. En type har Ethernet grensesnitt og plasseres foran webtjeneren som en egen tjener. Andre er innstikksmoduler i tjeneren i form av PCI eller SCSI moduler. De kan være som rene akseleratorer eller ha ekstra sikkerhetsfunksjoner. Sikkerhetsfunksjoner i kryptografiske moduler er standardisert i FIPS PUB 140-2 [9].

SSL/TLS har mulighet for gjenbruk av PKI nøklene via Session ID tracking. Denne mekanismen vil radikalt øke ytelsen mhp. hvor mange sesjonsetableringer tjeneren tåler så sørg for at den er påsatt.

## 4 Web tjenere

Ser en på de 20 mest kritisk sårbarhetene [10] (publiseres av FBI og Sans Institute) som rapporteres på Internet kan de grovt deles i to kategorier nemlig feilkonfigurering og ikke oppdatert programvare.

Før hackerne går til angrep går de stort sett gjennom tre trinn:

1. Footprinting
  - Samle informasjon om målet. Svært mye er åpnet tilgjengelig på Internett
  - Domenenavn, IP adresser, fysisk adresse, kontaktpersoner etc.
2. Scanning
  - Finner ut hvilke tjenester som er åpne
  - Finner ut hvilke operativsystemer som kjører (versjoner)
3. Enumeration
  - Kobler seg opp for å finne:
    - Brukernavn/passord
    - Nettverkstopologi
    - Versjoner (leter etter kjente hull)

Med denne informasjonen kan angrepet planlegges.

### 4.1 Microsoft Internet Information Server (IIS)

IIS har hatt sin dose med feil og patcher og stadig nye kommer til. Det er derfor svært viktig og regelmessig sjekke systemet mot nye patcher fra Microsoft. De har et verktøy som heter HFNetChk [11] for dette.

Ofte har det vist seg at IIS reagerer feil på uventede URLer av forskjellig slag. Det er et par ting en kan gjøre for å minimalisere problemer relatert til dette. Det første er å installere IIS Lockdown Tool [11]. Dette verktøyet vil også installere et annet verktøy nemlig URLScan. URLScan sjekker alle innkomne URLer mot konfigurasjonen for å se hva den skal slippe igjennom til IIS og hva den skal stoppe. Dagens versjon installerer URLScan versjon 2.0. Hent derfor ned og installer versjon 2.5 URLScan-SRP. SRP versjonen har tettere sikkerhets settinger enn baseline. En kan oppleve at

web applikasjonen ikke virker etter dette. Sjekk da loggen for å se hva den trigger på. Så kan du gjøre de nødvendige endringene i konfigurasjonsfila for å åpne for det som stoppes. URLScan kan også endre ServerToken. En kan f.eks legge inn "AlternateServerName=Webserver 1.0". Det anbefales også å partisjonere disken og legge webrota på en annen disk enn C: der system må ligge.

Bufferoverflow har vært en annen gjenganger i IIS. En server som har alle oppdaterte patcher er medisinen her. En annen gjenganger er eksempel applikasjoner og script som er en del av applikasjonen. De ligger f.eks. på %wwwroot%/scripts, få de vekk.

Microsoft har flere verktøy og tips på [11] så bruk de, de er gratis.

## 4.2 Apache

Apache [12] er verdens mest utbredte web tjener og den er gratis. Den finnes for mange plattformer også Windows. Den har hatt sine feil den også men ikke på langt nær så mange som IIS.

Det er et par ting en bør gjøre (se også [13]):

- Som for IIS gjelder det å ha oppdatert versjon av både Apache og OSet en kjører på.
- Ikke kjør serveren som root, og kjør den heller ikke som nobody (brukes til å mappe root over NFS i noen tilfeller).
- Fjern alt eksempel innhold (html, cgi etc.) som er noe av det første hackerne ser etter.
- Endre ServerTokens fra Full til ProductOnly. Dette vil fjerne versjonsnummeret men ikke Apache i tokenet. For å endre tokenet totalt må en recompile.
- Apache kommer med en mengde moduler. Installer bare de modulene som er nødvendig.
- Pass på at IncludesNoExec direktivet er med
- Filter funksjoner er det mange av. mod\_rewrite har mange funksjoner men kan brukes til å filtrere URLer. F.eks. vil RewriteRule \.\/|\$|% / [F,NE] kunne fjerne en del potensielle problemer.
- Slå av stack eksekvering (hvis mulig):
  - Solaris: set noexec\_user\_stack=1, set noexec\_user\_stack\_log=1
  - Linux: <http://www.openwall.com/linux/> (kjerne patch)
- Vurder å kjøre serveren i chroot() (det har sine sider) [14], [15].

En del kan gjøres gjennom konfigurasjon. Oppdatert programvare og fjerning av eksempel innhold er nok det viktigste.

## 5 Verktøy

Det er en del verktøy som kan sjekke installasjonen for vanlige feil. Det mest generelle verktøyet er Nessus [16] som er en sårbarhetsscanner. Den går gjennom alle portene, sjekker hvilke tjenester som kjører og melder fra om kjente sikkerhetshull. Den kjører på Posix systemer (Solaris, Linux etc.) men det finnes en Win 32 klient som kan brukes til å administrere Nessus installasjonen og generere rapporter. Dette er et generelt nyttig verktøy som kan brukes på mye mer enn webtjenere.

Nikto [17] er en scanner spesielt for web som bruker en modul som heter LibWhisker. Den dekker mange forskjellige sikkerhetsproblemer på mange forskjellige webtjenere.

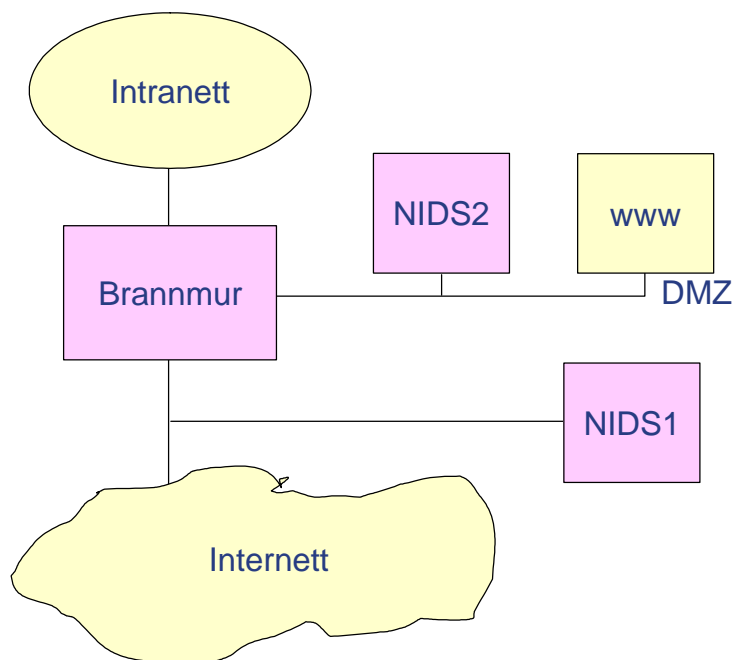
Open Web Application Security Project (OWASP) [18] er et Open Source prosjekt for å utvikle både verktøy og kunnskapsbase for å hjelpe til å sikre web applikasjoner. WebScarab [19] er en web sikkerhetsscanner skrevet i Java som er i en tidlig utgave i skrivende stund.

Talisker har en oversikt over mange scannere både freeware og kommersielle [20]. The World Wide Web Security FAQ er også en god kilde til informasjon om web sikkerhet [21].

## 6 IDS etc.

Det finnes mange type IDS (Intrusion Detection System) systemer. Som hovedklasser kan nevnes nettverk IDS (NIDS) og host IDS (HIDS). NIDS sniffer pakker på nettet og analyserer de for å se om den finner mistenkelige ting. HIDS er installert på den verten en vil overvåke. Den sjekker systemlog og annet for å lete etter mistenkelig aktivitet. Bare HIDS kan detektere angrep som skjer oppå en SSL forbindelse for eksempel ved hjelp av stunnel [22]. En annen inndeling av IDS er i signatur basert og anomali basert. Signatur baserte IDSer ser etter bestemt mønster nesten som en virus scanner mens anomali baserte ser etter endringer i mønster etc.

Det finnes mange måter å plassere NIDS på avhengig av hva en ønsker å oppnå. To



Figur 3: Plassering av NIDS

plasseringer er vist i Figur 3. Ved å plassere NIDS foran brannmuren som vist med NIDS1 vil en monitorere all trafikk. Dette kan være ønskelig men en skal være oppmerksom på at dette vil skape mye støy. Hvis en spesifikt vil overvåke webtjeneren kan den plasseres på DMZ som vist med NIDS2. Brannmuren vil filtrere vekk all trafikk som ikke er konfigurert i brannmuren at den skal til www tjeneren. Dette vil filtrere vekk mye støy men det kan også bli svært vanskelig å oppdage de initielle manøvrene hackeren gjør før de setter inn støtet. Da tenker jeg på de innledende øvelsene som beskrevet i

kapittel 4. Hva som er hensiktsmessig må bli en vurdering i hvert tilfelle. Det er avhengig av både security policy, hva en er bekymret for og hvor mye ressurser en har. Det kan også være et alternativ å kjøpe NIDS overvåknings tjenester. Da vil NIDS stå ved tjeneren en ønsker å overvåke mens kontroll og overvåkning skjer hos

tjenesteleverandøren. NIDS må tunes kontinuerlig for å holde støy nede og å kunne oppdage nye angrepstyper.

Snort [23] er en hovedsaklig signaturbasert NIDS som er svært utbredt og den er gratis. Snort er tilgjengelig for Linux og Win 32 og har et stort antall signaturer som dekker mange typer angrep. Det finnes mange tilleggssystemer til Snort som SnortCenter (sentral administrasjon), SnortSnarf (genererer html rapporter) og SnortSam (blokkerer IP adresser).

Et annet Open Source IDS system som er interessant er Prelude [24]. Prelude er et såkalt Hybrid IDS dvs. det er både NIDS og HIDS ved at det kan analysere syslog samlet inn fra mange kilder i tillegg til å ha NIDS.

En type systemer er bare satt opp for å trekke til seg hackere og logge aktiviteten deres. Disse kalles honeypots. Dette begynte med Deception Tollkit utviklet av Fred Cohen [25] i 1997. Siden er det kommet til mange prosjekter. Tracking Hackers [26] har en fin oversikt over slike systemer. Et interessant prosjekt er Bait-n-Switch [27]. Dette tar trafikk som ikke er lovlig trafikk og switcher den over til en honeypot for analyse.

Hogwash [28] er et såkalt Intrusion Prevention System (IPS). Dette er ikke en passiv sniffer men et system som trafikken passerer gjennom. Det inneholder en NIDS modul som analyserer trafikken og en modul som blokkerer uønsket trafikk.

Codeseeker [29] sitter på selve web serveren og kan være både HIDS og HIPS. En proxy versjon er under utvikling.

Det kan være på sin plass å advare litt mot aktive teknikker som IDS med strikeback, IPS etc. da de må konfigureres varsomt for å unngå at en lager tjenestenektangrep på seg selv. En må vite hva en gjør når en bruker slike teknikker.

## 7 Tjenestenekt, (D)DoS

Tjenestenektangrep (DoS, Denial of Service) har grepet om seg i de senere årene. I mange sammenhenger kan det være viktig at bedriften er kontinuerlig på nettet. Et angrep som bringer bedriften av nettet over tid kan derfor være svært skadelig. Noen av disse angrepene utnytter svakheter i systemet slik at det krasjer (oppdatert programvare!!!). Mange av angrepene er distribuerte, DDoS, ved at et stort antall maskiner er blitt penetrert og blitt brukt som verktøy i et distribuert angrep mot et mål. Slike angrep går stort sett ut på å fylle all båndbredden mot bedriften med trafikk slik at nyttetraffic ikke kommer gjennom. En del av dette kan elimineres eller begrenses ved at ISP'en implementerer gode prosedyrer for filtrering. Best Current Practice nr 38 [30] beskriver en del tiltak som bør tas.

Det en bør gjøre er å sørge for at ens eget nettsted er konfigurert i henhold til BCP38 og holde all programvare oppdatert slik at en ikke selv blir en av de som blir brukt som verktøy mot andre. Ofte kommer slik trafikk fra ikke tildelte nett. Jeg har en oversikt over slike nett [31]. Hvis en ser trafikk (på IDSen eller brannmurlogg etc.) fra disse adressene så følger ikke ISP'en BCP38. Det anbefales da noen sure e-post til [abuse@ispen.din](mailto:abuse@ispen.din) som forlanger at de får huset sitt i orden (sørg for å ha orden hjemme først).

En kan også samarbeide med ISPen å filtrere visse trafikktyper etter båndbredde. Det er ingen fornuftig grunn til at Ping trafikk (eller annen ICMP trafikk) skal kunne ta 100% av båndbredden inn til bedriften. 5% burde holde lenge.

## 8 Avslutning

Når en har vært gjennom alle disse tiltakene skulle en tro at en kan sove godt om natta. De fleste websteder har aktivt innhold med oppslag i databaser, søkeverktøy, skjema for e-postrespons osv. Det åpner opp for nye feil og nye muligheter for hackerne til å trenge seg inn. Sov godt.

## 9 Referanser

- [1] <http://www.rfc-editor.org/rfc/rfc2923.txt>
- [2] <http://inner.net/opie>
- [3] <http://www.rfc-editor.org/rfc/rfc2246.txt>
- [4] <http://monkey.org/~dugsong/dsniff/>
- [5] <http://www.rfc-editor.org/rfc/rfc3377.txt>
- [6] <http://www.rfc-editor.org/rfc/rfc2865.txt>
- [7] <http://www.iddefense.com/idpapers/SessionIDs.pdf>
- [8] Utimaco: KeyGrab T00, The search for keys continues...
- [9] <http://csrc.nist.gov/cryptval/140-2.htm>
- [10] <http://www.sans.org/top20/>
- [11] <http://www.microsoft.com/technet/security/tools/tools.asp>
- [12] <http://www.apache.org/>
- [13] [http://httpd.apache.org/docs-2.0/misc/security\\_tips.html](http://httpd.apache.org/docs-2.0/misc/security_tips.html)
- [14] <http://www.devet.org/apache/chroot/>
- [15] <http://penguin.epfl.ch/chroot.html>
- [16] <http://www.nessus.org/>
- [17] <http://www.cirt.net/code/nikto.shtml>
- [18] <http://www.owasp.org/>
- [19] <http://www.owasp.org/webscarab/>
- [20] <http://www.networkintrusion.co.uk/>
- [21] <http://www.w3.org/Security/Faq/>
- [22] <http://www.stunnel.org/>
- [23] <http://www.snort.org/>
- [24] <http://www.prelude-ids.org/>
- [25] <http://www.all.net/>
- [26] <http://www.tracking-hackers.com/>
- [27] <http://baitnswitch.sourceforge.net/>
- [28] <http://hogwash.sourceforge.net/>
- [29] <http://www.owasp.org/codeseeker/>
- [30] <http://www.rfc-editor.org/rfc/bcp/bcp38.txt>
- [31] <http://www.presttun.org/kare/network/>