BACHELOROPPGAVE:

KVALITETSSIKRING AV EXPLOITS FOR PENETRASJONSTESTING

FORFATTER(E): BJØRNAR PRESTAASEN
MATHIAS BJERKE
TRULS HAGEN
GUDMUND NORDSTRØM

Dato: 20.05.2009

SAMMENDRAG AV BACHELOROPPGAVEN

Tittel:	Kvalitetssikring av exploits for bruk i penetrasjonstesting	Nr. : 7 Dato : 20-05-2009
Deltaker(e):	Truls Hagen Bjørnar Prestaasen Gudmund Nordstrøm Mathias Bjerke	
Veileder(e):	André Årnes, Høgskolen i Gjøvik	
Oppdragsgiver:	Ernst & Young	
Kontaktperson:	Eirik Thormodsrud	
Stikkord (4 stk)	Penetrasjonstesting, exploits, ondsinnet kode, kvalitetssikring	
Antall sider:172	Antall bilag:15	Tilgjengelighet (åpen/konfidensiell) :Åpen

På daglig basis blir det innrapportert nye sårbarheter i ulike typer programvare og tjenester. Samtidig blir det utviklet kode (exploits) for å utnytte disse, med den hensikt å kompromittere sårbare systemer. Denne informasjonen sprer seg raskt, og for å holde seg oppdatert kreves det mer og mer fra de som har ansvar for IT-sikkerheten.

For å beskytte seg mot dette trusselbilde er man nødt til å tenke på samme måte som en angriper. En metode for å beskytte seg kan være å utføre en penetrasjonstest for å kartlegge eventuelle sikkerhetshull. Penetrasjonstesting er prosessen der man benytter kvalifisert personell til å simulere reelle angrep mot datasystemer eller nettverk, med det formål å evaluere og forbedre den eksisterende datasikkerheten. Ved å bruke ulike typer verktøy som exploits og annen potensielt skadelig programvare sørger man for at testen blir så reell og troverdig som mulig.

For å kunne benytte exploits eller potensielt skadelig programvare, må man være helt sikker på at koden oppfører seg som forventet, og ikke resulterer i uønskede bivirkninger eller i verste fall ødeleggelse av kundens produksjonssystemer.

Dette prosjektet går ut på å lage et konseptuelt rammeverk for å forsikre seg om at nettopp dette ikke skjer. Vi har sett på ulike typer metodikker og rammeverk for hvordan man kan kvalitetssikre nye exploits. I tillegg er det blitt laget et konseptuelt rammeverk for gjennomføring og testing av exploits hvor vi har testet to exploits for å verifisere systemet.

For å forenkle kvalitetssikringen så har vi valgt å fokusere på forskjellige måter å automatisere systemet. Dette innebærer blant annet logging av aktivitet fra flere forskjellige kilder, filtrering av data og dynamisk presentasjon av disse på flere nivåer.

Bjørnar Prestaasen, Gudmund Nordstrøm,
Mathias Bjerke og Truls Hagen

2009

Kvalitetssikring av Exploits for penetrasjonstesting

Forord

Kvalitetssikring av exploits var en av oppgavene Høgskolen i Gjøvik lanserte høsten 2008. Som informasjonssikkerhetsstudenter hadde vi ønske om å gjøre noe rettet mot sikkerhetstesting. Interessen for oppgaven kom når vi så at den omhandlet exploits og kvalitetssikring av disse. Å få arbeide med å lage et testmiljø for kvalitetssikring av exploits, er noe som er lærerikt både med tanke på hvordan datamaskiner oppfører seg, og hvordan den kan overvåkes. At Ernst & Young som driver med IT-revisjon kan ha behov for et slikt system, er i seg selv en inspirasjon.

Oppgaven var utfordrende med tanke på å få på plass metodikker og det konseptuelle rammeverket før vi kunne begynne og teste exploits. Å beskrive noe så grundig før vi laget systemet er utfordrende i seg selv, men motivasjonen med og kunne gjennomføre tester i et sikkert miljø og overvåke endringer var spennende.

Vi ønsker å gi en stor takk til vår veileder gjennom prosjektet, André Årnes. Uten hans faglige kompetanse på området og fokus på fremdrift i riktig retning hadde vi ikke kommet dit vi er i dag.

Vi vil også gi en stor takk til Eirik Thormodsrud og Hasse Kristiansen fra Ernst & Young for ideen til en spennende og utfordrene oppgave. Vi vil også takke de for imøtekommenheten og samarbeidsviljen for å få frem prosjektet og ideen.

Gjøvik, 25. mai 2009

Bjørnar Prestaasen	Gudmund Nordstrøm	Mathias Bjerke	Truls Hagen

Figurer

Tabeller

Definisjoner

Ord	**Forklaring**
API	Application Programming Interface
CSV	Comma-separated values. Filformat hvor alle kolonner er separert med et komma
Dynamisk analyse	Detektere og analysere endringer i sanntid
Exploit	Kode brukt for å utnytte sårbarheter i programvare
Hendelse	En lest fil, skrevet fil, nettverkspakke, etc
Obfuskert kode	Måte å prøve å forvirre lesere av kildekode slik at det er vanskelig å se hva et program gjør
Overflateanalyse	Før- og etteranalyse som sammenlignes for å finne endringer
Payload	Kode som blir injisert eller igangsatt i det en exploit er gjennomført
POSIX	Portable Operating System Interface. Standard som definerer grensesnittet mellom programmer og UNIX operativsystemer
Ressurser	En dataressurs. For eksempel fil, prosess, nettverk, register eller minne.
Rootkits	Et sett med programmer som skjuler at et system har blitt tatt over
Snapshot	Speilbilde/kopi av datamaskinens tilstand
Statisk analyse	Analyse av et system eller kode som ikke kjører
Trojaner	Program som utgir seg for å være noe annet en det egentlig er
Virus	Ondsinnet programvare som kopierer seg selv inn i filer eller datamaskinens oppstartsektorer. Krever at brukeren eksekverer koden.
Whitebox	Testing hvor man har tilgang på kildekode og prøver å teste ut ifra den

Innhold

1 Innledning

Dagens trusselbilde er dynamisk. Hver dag blir det rapportert nye sårbarheter[1][2] samtidig som kode (exploits) for å utnytte disse blir tilgjengeliggjort[3]. For å være forbredt på morgendagens trusler er man derfor nødt til å tenke proaktivt, noe som innebærer at man til tider må tenke på samme måte som en angriper. Et kjent uttrykk innen informasjonssikkerhet sier at "en angriper trenger kun å finne én sårbarhet for å kompromittere et system, mens en sikkerhetsansvarlig må finne alle". Dette kan høres urettferdig ut, men slik er det blitt i dag. En komplett sikkerhetstest med sårbarhetsanalyse og penetrasjonstesting kan derfor være ønskelig for å teste et system, før en eventuell angriper gjør det. Ved å utføre en sårbarhetsanalyse får man en oversikt over potensielle sårbarheter, og for å verifisere disse brukes en penetrasjonstest.

Ved gjennomføring av en fullstendig sikkerhetstest er det ønskelig å utnytte kartlagte sårbarheter for å se hvilke skader et ekte angrep kan gi. Det er også ønskelig at en penetrasjonstest skal være så reell som mulig, noe som tilsier at sikkerhetstesterne til tider må bruke nye (zero-day) exploits som blir tilgjengeliggjort.

Flere og flere bedrifter velger å utføre en penetrasjonstest mot sine systemer, men for mange blir bruk av nye exploits utelatt med frykt for at det kan skape uønskede bivirkninger i eventuelle produksjonssystemer. Usikkerheten rundt bivirkningene til angrepsverktøy/exploits og hva dette kan gjøre med målsystemet kan derfor påvirke påliteligheten til testen. En annen faktor som ytterligere forsterker denne usikkerheten er at man ikke kan være sikker på at den som har skrevet exploiten ikke har obfuskert koden og med vilje har plantet inn ondsinnet kode. Skulle man komme til å benytte en exploit/payload av denne typen risikerer man å skape brudd på konfidensialitet, integritet eller tilgjengelighet til testobjektet. Denne usikkerheten deles også av vår eksterne veileder, Ernst & Young (EY), og ønsker derfor et rammeverk hvor man på en effektiv måte kan kvalitetssikre exploits/payloads før de benyttes som en del av en sikkerhetstest.

Dette kapittelet omhandler informasjon om prosjektoppgaven, hvem prosjektet er skrevet for og hvem det er skrevet av. I tillegg beskrives organiseringen av rapporten og hvordan gruppen har jobbet.

1.1 Målgruppe

Målgruppen for rapporten er først og fremst EY, men vil i tillegg være aktuell for studenter og ansatte ved Høgskolen i Gjøvik, samt andre som interesserer seg for IT-sikkerhet, dataetterforskning eller analyse.

1.2 Formål

EY ønsker en effektiv og etterprøvbar kvalitetssikring av exploits til bruk i penetrasjonstesting. Det å kunne bruke exploits på en sikker måte krever at man vet eksakt hvordan målsystemet blir påvirket. For å unngå å lese kildekode linje for linje, ønsker EY et rammeverk hvor man kan forenkle denne prosessen.

Målet med prosjektet er å utvikle et konseptuelt rammeverk for overvåkning av exploits i et sikkert miljø. Videre skal det genereres rapporter fra overvåkningen som viser hvilke endringer som er påført målsystemet under testingen. Ved å ta i bruk dette konseptet, ser vi for oss at EY enkelt vil kunne ta i bruk et større utvalg av exploits for testing ute hos sine kunder.

Prosjektets effektmål er å gi EY et hjelpemiddel de kan benytte for å teste og kvalitetssikre exploits. Dette vil gi EY mulighet for å utføre mer omfattende tester og øke kvaliteten på sine tjenester.

Prosjektets resultatmål er at rammeverket som skal utarbeides må være så generelt at det enkelt kan overføres til forskjellige plattformer. Selve prosjektrapporten vil være dokumentasjonen for både rammeverket og implementasjonen. Brukerveiledning og policy skal utarbeides for at brukerne skal kunne bruke rammeverket på en trygg og effektiv måte.

Læringsmålene for prosjektet er at gruppen skal få en dypere forståelse rundt analyse av exploits både i teori og praksis. Gruppen vil få erfaring med testing i virtuelt miljø, bruk av ulike verktøy for overvåkning av ressurser, samt bruk av forskjellige prosjektstyringsverktøy og prosjekt som arbeidsform.

1.3 Motivasjon

Motivasjonen for prosjektoppgaven er ønsket om at EY raskt skal kunne ta i bruk nye exploits ved penetrasjonstestingen hos sine kunder, og ytterligere forsterke påliteligheten til egne tester. Det å få innsikt i hvordan exploits oppfører seg i forhold til operativsystemet vil være veldig lærerikt, og det å få god innsikt i virtualisering og testing i sikre miljøer er også gode erfaringer å ta med seg videre.

1.4 Gruppens bakgrunn

Gruppens medlemmer er alle fra siste året i Bachelor i Informasjonssikkerhet. To av medlemmene har bakgrunn fra elektronikkfag, mens de to siste henholdsvis kommer fra idrettslinje og IKT-driftsfag. Felles for alle er et ønske om å utdanne seg innen IT med et målrettet fokus på sikkerhet.

Bjørnar Prestaasen
Kontaktperson

Mob: 980 73 176
bjprest (a) gmail.com

Bjørnar er 24 år gammel og etter grunnskolen gikk han tre år på idrettslinja ved Stange VGS. Etter ett år i forsvaret ved HTTS Kampeskadronen gikk han grunnfag idrett ved Høgskolen i Hedmark før han startet på Bachelor i Informasjonssikkerhet ved Høgskolen i Gjøvik.

Gudmund Nordstrøm
Dokumentansvarlig

Mob: 958 63 877
gudmund84 (a) gmail.com

Gudmund har fagbrev som IKT-driftsoperatør etter to år som lærling ved IKT-faggruppen i Hamar kommune. Oppgavene i Hamar kommune var hovedsaklig drift og brukerstøtte av Novell-nettverk og Windows 2000. Gudmund har også gjennomført førstegangstjeneste hos Forsvarets Sikkerhetsavdeling hvor oppgavene blant annet var administrasjon og drift av informasjonssystemer og vedlikehold av teknisk lab ved Forsvarets Ingeniørhøgskole.

Mathias Bjerke
Informasjonsansvarlig

Mob: 971 41 810
mathbje (a) gmail.com

Mathias har flere års erfaring som programutvikler. Det siste året har han jobbet med utvikling av EDI-løsninger i eget selskap. Mathias har fagbrev innen Serviceelektronikk fra Gjøvik VGS og fullfører nå en bachelor i Informasjonssikkerhet ved Høgskolen i Gjøvik.

Truls Hagen
Prosjektleder

Mob: 922 54 312
truls.hagen (a) gmail.com

Truls har bakgrunn fra elektronikk på Gjøvik VGS og fullfører nå en Bachelor i Informasjonssikkerhet ved Høgskolen i Gjøvik. Han jobber i dag med smartkortteknologi og sikker identifisering på nett. Privat har Truls flere års erfaring med bruk av Linux-systemer, med en spesiell interesse for drifting og sikring av disse.

1.5 Arbeidsmetoder og prosjektstyring

Som arbeidsmetode ble det benyttet gruppearbeid. Interne oppgaver ble fordelt tidlig slik at alle gruppemedlemmene hadde klare arbeidsoppgaver under hele prosjektet. Statusmøter ble avholdt hver uke, og med jevne mellomrom ble disse holdt sammen med veileder André Årnes. Møtene ble gjort over telefon eller sammen på HiG. To større møter ble holdt sammen med veileder og EY. Disse ble holdt både i Gjøvik og i Oslo,

henholdsvis 20. februar og 28. april.

Under hele prosjektet ble det benyttet et prosjektstyringsverktøy som heter Trac. En av de største fordelene til dette verktøyet er dens støtte mot versjonshåndteringssystemet Subversion. I Trac finnes det også mange hjelpemidler for å styre prosjekter, der bruk av wiki og et ticketsystem ble mest brukt i dette prosjektet. Disse funksjonene var til stor hjelp for å holde oversikt over fremdriften i prosjektet. Møtereferat og loggbok for hvert gruppemedlem ble lagt inn på wikisiden.

1.6 Oppgavebeskrivelse

Ved penetrasjonstesting er det ofte nødvendig/ønskelig å benytte seg av exploits eller annen potensielt skadelig kode for å utføre en mer grundig/reell test av målsystemet. Koden som skal kjøres må vise en tilstrekkelig grad av sikkerhet, slik at man kan vite at koden kun gjør det den skal. Det er derfor ønskelig å kunne teste koden i et sikkert miljø der man kan overvåke endringer i målsystemet og verifisere kodens funksjon, uten nødvendigvis å måtte lese gjennom hele kildekoden. Oppgaven vil derfor gå ut på å utvikle et konsept og en løsning som tar for seg overvåkning av potensielt skadelig programvare i et kontrollert testmiljø. Dette testmiljøet vil bli kontrollert og overvåket på to måter: i sanntid og ved analyse før og etter utført test. Resultatet skal til slutt sammenstilles for å finne ut hvordan gitte ressurser hos målsystemet har blitt påvirket av koden.

Koden som testes skal kunne bli eksekvert enten lokalt eller fra en annen maskin via nettverket. Når det gjelder målsystemet skal det fokuseres på følgende ressurser, i prioritert rekkefølge:

1. Endringer i filer
2. Prosesser (startede og avsluttede)
3. Nettverk (tilstand og trafikk)
4. Endringer i Windows register
5. Endringer i internminnet

Målet er å få en verifikasjon på at exploiten gjør det den skal, og ikke påvirker andre prosesser ved for eksempel å innføre rootkits eller andre uønskede programmer. Det skal videre utarbeides forslag til brukerveiledning og policy for hvordan tester skal gjennomføres og rapporteres. Dokumentasjonen skal være generell nok til å kunne benyttes på flere systemer. Den bør dekke hvordan systemet skal settes opp, benyttes og tilbakestilles for å kunne ha et rent testmiljø til enhver tid.

Oppgaven vil i hovedsak gå ut på å utvikle et generelt konsept og rammeverk med tanke på muligheter for videreføring til flere plattformer. På grunnlag av dette bør det vises produktuavhengighet i så stor grad som mulig. I diskusjon med EY har vi kommet fram til at testingen (eksperiment/proof of concept) først og fremst skal skje på Windows XP-plattformen. I hovedsak på grunn av utbredelse og antall tilgjengelige exploits.

1.7 Faglige bidrag

- Konseptuelt rammeverk for testing og kvalitetssikring av exploits for bruk i penetrasjonstesting
- En applikasjon som automatiserer datainnsamlingen, filtrering og presentasjon av resultatene
- Eksperimenter med reelle exploits som verifiserer overvåkningen av ressursene

1.8 Ansvarsfordeling og roller

Tildeling av roller og medfølgende arbeidsoppgaver innad i gruppen ser vi på som en absolutt nødvendighet for å skape struktur og stabilitet for prosjektet og gruppearbeidet. Vi har derfor valgt å dele gruppens medlemmer på fire ansvarsområder. Ansvarsområdene er prosjektleder, kontaktperson, informasjonsansvarlig og dokumentansvarlig.

1.8.1 Gruppens ansvarsfordeling

Truls Hagen:	Prosjektleder. Ansvar for fremdrift og fordeling av oppgaver.
Gudmund Nordstrøm:	Dokumentansvarlig. Ansvar for fremdrift og ferdigstilling av dokumenter.
Bjørnar Prestaasen:	Kontaktperson og sekretær. Ansvar for å distribuere informasjon.
Mathias Bjerke:	Informasjonsansvarlig. Ansvar for webside og prosjektstyringsverktøy.

1.8.2 Eksterne roller

Ekstern veileder:	Ernst & Young AS
Kontaktperson:	Eirik Thormodsrud
Veileder:	André Årnes

1.9 Begrensning av omfang

Å teste en ondsinnet kode er ikke en ny metode som har kommet i de senere årene, da det finnes flere forskjellige verktøy og rammeverk som spesialiserer seg på nettopp dette: å detektere og fjerne uønsket kode. De fleste av disse verktøyene tar for seg testing av malware som virus, ormer, trojanere etc. I prosjektet skal vi utarbeide et konseptuelt rammeverk for å kunne kvalitetssikre exploits i et testmiljø. Vi skal overvåke viktige ressurser for å detektere hva exploiten gjør med målsystemet. Dette for å kunne velge

ut exploits som kan tas inn i penetrasjonstesting, med bedre oversikt over virkemåten til exploiten. Det å automatisere denne prosessen så langt det er mulig gjør at menneskelige feil unngås, noe som vil bli vektlagt gjennom hele rapporten.

Ressursene som overvåkes er som nevnt tidligere prioritert i følgende rekkefølge: filer, prosesser, nettverk, register og minne. Mye av grunnlaget for denne prioriteringen er basert på forholdet mellom arbeidsmengde og tid, samt hvordan vi best kan tilegne oss erfaringer på disse områdene.

I prosjektet har vi valgt å bruke eksisterende verktøy til datainnsamlingen siden det finnes mange gode verktøy som allerede har blitt testet og er godt dokumentert. I analysedelen har vi utviklet egne verktøy fordi det er få som passet til vårt formål. Testmiljøet har blitt satt opp virtuelt, med de tilpasninger som er nødvendige for å ha et identisk testmiljø for hver testcase.

Selve eksperimentet vil bestå av tre uavhengige testcaser. Det første testcaset er en fiktiv "exploit", som vil bli brukt for å kontrollere testmiljøets funksjonalitet, og for å få testmiljøet til å fungere optimalt. Når vi har fått god innsikt på disse områdene, vil det testes en kjente og dokumentert exploit. Det tredje testcaset vil bestå av en exploit som er mer kompleks og gjør flere endringer på systemet. Dette gjøres for å sammenligne resultatene opp i mot dokumentasjonen på exploitene, eller andre som har testet koden.

1.10 Organisering av rapport

Rapporten består av 9 kapitler. De tre første kapitlene omhandler informasjon rundt gruppen og rapporten, fagteori på området og hvilke metoder vi har valgt å bruke. De påfølgende kapitlene omhandler rammeverket, testmiljøet og resultater av testene vi har utført. Til slutt har vi en oppsummering og en konklusjon.

1. **Innledning**
   Innledning til prosjektoppgaven
2. **Bakgrunn og fagteori**
   Relatert arbeid og teori rundt områdene prosjektoppgaven dekker
3. **Metodikk**
   I dette kapittelet går vi gjennom forskjellige metoder og teknikker som kan brukes i prosjektet
4. **Konseptuelt rammeverk og testverktøy**
   Her kommer ideen som vi skal utvikle og hvilke verktøy som kan brukes
5. **Eksperiment og testmiljø**
   Oppsett av testmiljø og beskrivelse av de tre eksperimentene vi skal utføre
6. **Resultat**
   Resultatene fra eksperimentene

7. **Diskusjon og Konklusjon**
   Oppsummering og evaluering av resultatene
8. **Referanser**
   Inneholder hvilke kilder vi brukte i prosjektet
9. **Vedlegg**
   Vedleggene inneholder forskjellige dokumenter som vi har valgt å flytte ut fra selve rapporten

Appendix A Inneholder koden til autoscript.sh
Appendix B Loggagent. Start.bat og Stop.bat
Appendix C Fiktiv exploit
Appendix D Kode for monitor.cs
Appendix E Kode fra Webside for presentasjon
Appendix F Rapport for fiktiv exploit
Appendix G Rapport for kjent exploit
Appendix H Rapport for kompleks exploit
Appendix I Forslag til policy
Appendix J Forslag til brukerveiledning
Appendix K Prosjektavtale
Appendix L Forprosjektrapport
Appendix M Reellt gantt-skjema
Appendix N Møtereferat
Appendix O Sammendrag prosjektdagbok

2 Bakgrunn

For å gjennomføre prosjektet på en tilfredsstillende måte er det nødvendig å se på relatert arbeid. Som nevnt tidligere er ikke hensikten å finne opp hjulet på nytt, men heller å dra nytte av det som er gjort tidligere, og jobbe videre fra dette. Som bakgrunnsstoff har gruppen derfor sett på aktuelle bøker og prosjekter relatert til exploits og malware, samt noen exploit-rammeverk. Vi har også sett på grunnleggende fagteori rundt malware, exploits og penetrasjonstesting, og skal i dette kapittelet vise hvordan dette henger sammen med oppgaven.

Gruppen har måttet tilegne seg kunnskap rundt virtualisering, analysemetodikker, API-hooking og filsystemer. Fagene som gruppen har gjennomført på HiG, eksempelvis penetrasjonstesting, operativsystemer, nettverksikkerhet og datamaskinarkitektur, har alle gitt et godt grunnlag for mye av prosjektets innhold.

Når vi har sett på tidligere arbeid finnes det få åpne prosjekter som spesifikt omhandler analyse og kvalitetssikring av exploits. Det er derimot skrevet om analyse av malware [4] og testing av antivirusverktøy [5]. Det finnes også nettsider som tilbyr online analysering av malware [6]. Nedenfor vises det til fagteori om de mest berørte temaene i rapporten.

2.1 Exploits

En exploit er programkode spesielt laget for å utnytte kodefeil, systemfeil eller sårbarheter som forårsaker uventede resultater i maskinvare og/eller programvare. Målet for denne programkoden kan være å eskalere rettigheter til seg selv, ta full kontroll over målet, utføre tjenestenektangrep eller eksekvering av egen kode.

Exploits kan kategoriseres på flere måter. De mest kjente er hvordan en exploit *kontakter* målsystemet. En ”remote” exploit tar kontakt ved bruk av nettverket, mens en ”local” exploit befinner seg på den samme maskinen. Felles for begge er at de utnytter sårbarheter i programmer og tjenester for eksempelvis å få eskalere rettigheter eller eksekvere egen kode (payload). En annen kategorisering er hvordan exploiten *angriper* målsystemet. Exploits som utfører tjenestenektsangrep over nettverk og manuell injeksjon av ondsinnet kode er eksempler på en slik kategorisering.

Problemet vi skal ta for oss i denne rapporten, er at en exploit i utgangspunktet kan være designet for å inneholde uønsket kode eller gjøre mer skade på målsystemet eller miljøet rundt enn det den utgir seg for. Det kan også være at exploiten er skrevet for dårlig eller at den i noen sammenhenger vil være ustabil og ødelegge tjenester på målmaskinen, eller målmaskinen selv. Et eksempel på dette kan være en exploit som utgir seg for utnytte en sårbarhet i Apache[1] og returnere et supershell[2], men i stedet henter ut personinformasjon fra både angriper og målmaskin og for deretter å sende dette videre til en tredjepart.

[1] Web-tjener for UNIX - http://www.apache.org/

[2] shell med administrative rettigheter

Vi har også sett på exploit-rammeverket Metasploit Framework[7]. Rammeverket er utbredt og inneholder exploits for flere plattformer, samt en mengde payloads til disse. Eksempler på payloads kan være "Reverse Shell-connections" og Reverse VNC-connection[3]. Rammeverket finnes i flere utgaver for ulike typer plattformer, og kan også brukes grafisk med et svært intuitivt grensesnitt. Programmet blir stadig oppdatert med nye typer exploits.

2.2 Malware

Malware er en fellesbetegnelse for all ondsinnet kode. Ondsinnet kode kan for eksempel være virus, trojanere, rootkits eller programmer som logger tastetrykk. Ifølge F-secure ble det produsert like mye malware i 2007 som i de 20 foregående årene til sammen[8]. Denne trenden viser at produksjonen av malware har økt betraktelig den siste tiden og at fokus på it-sikkerhet blir mer og mer viktig for å kunne håndtere morgendagens trusler.

Et annet eksempel er Symantec sin rapport "Symantec Internet Security Threat Report: Trends for July–December 07"[9], som sier at det finnes indikasjoner på at det blir produsert like mye malware som det blir laget legitim software. Dette setter produksjonen av malware i perspektiv hvis man tenker over hvor mye legitim software det produseres pr. år.

2.3 Penetrasjonstesting

Penetrasjonstesting er en del av "etisk hacking", og er en metode tatt i bruk av sikkerhetseksperter for å avdekke svakheter og feilkonfigurasjoner i tjenester, operativsystemer, sikkerhetsbarrierer, nettverk og andre systemer tilknyttet virksomhetens IT-systemer. En del av prosessen går blant annet ut på å benytte ulike typer verktøy, deriblant exploits, i et forsøk på å bryte seg inn i systemet og simulere angrep. Etter fullført test lages det og fremstilles rapporter av eventuelle avdekkede sikkerhetshull og hva som kan gjøres for å tette disse.

Det finnes flere gode grunner for at bedrifter bør penetrasjonsteste egne systemer[10]. Først og fremst må systemene sikres for angrep som medfører finansielle tap. I tillegg kan penetrasjonsstesting avdekke sårbarheter slik at man kan jobbe proaktivt med systemene, for å forhindre uønskede situasjoner.

2.4 Digital dataetterforskning

Aldri har samfunnet vært mer avhengig av datamaskinen. Vi blir mer og mer digitaliserte, og minst mulig skal gjøres med penn og papir. Samtidig som antall brukere øker, er det også flere kriminelle som ser sitt snitt til å utnytte dette. Havner saken i rettsalen er man

[3] Programvare for å fjernstyre målmaskin grafisk

avhengig av at det er gjort en god dataetterforskning (eng.: digital forensics) for å sikre eventuelle bevis.

Dataetterforskning består i hovedsak av de tre grunnleggende stegene[11] innhenting, analyse og presentasjon.

1. **Innhenting**
   Denne fasen tar vare på tilstanden til et digitalt system slik at det i ettertid kan analyseres. Typiske eksempler er å lage et digitalt avtrykk av for eksempel lagringsmedia.

2. **Analyse**
   Her tar man i bruk dataene skaffet i forrige fase for å lete etter bevis som enten støtter eller motsier en teori. Det kan også være at man finner spor etter ødeleggelse av bevis, i den sammenheng at noen vil skjule sine spor. Eksempler kan være å gjennomgå filsystemer og/eller gjenopprette slettede data.

3. **Presentasjon**
   Den siste delen presenterer konklusjoner av eventuelle funn, og er som oftest ment for bruk i retten.

Stegene vist her er mye likt de stegene som vil brukes for kvalitetssikring i dette prosjektet. Det er derfor nærliggende å tenke at deler av rapporten også kan bli benyttet på dette området. Typiske deler kan være å finne ut hva et angrep/virus har gjort, reprodusere den samme metoden/angrepet i et sikkert miljø eller lære seg mønsteret/fremgangsmåten til angriperen. Viktige faktorer som da må ivaretas er opprettholdelse av integritet og etterrettelighet med tanke på gyldigheten til eventuelle bevis. Det er tidligere blitt laget flere ulike typer testmiljøer og metoder[12] for å studere angrep, og verktøy brukt innen rekonstruksjon i dataetterforskning. Virtual Security Testbed (ViSe)[13] er ett av disse, og er basert på et virtuelt testmiljø der det testes angrep mot de mest brukte operativsystemene.

2.5 Intrusion Detection System

Et IDS er kort fortalt et system med formål å monitorere, avdekke og i noen tilfeller avverge angrep eller oppførsel som ikke er legal/forventet. Et tradisjonelt IDS er plassert *bak* brannmuren, i den hensikt å analysere aktivitet innad i nettverket. Dette kan for eksempel være for å verifisere at trafikken som blir sluppet gjennom brannmuren faktisk er legitim.

Det finnes flere typer IDS, men de mest relevante innen dette området er host-, nettverk- og applikasjonsbaserte. Et hostbasert IDS (HIDS) er som navnet tilsier, installert på en host (vert) med formål å detektere endringer på akkurat denne enheten[4]. Nettverksbaserte IDS (NIDS) er strategisk plassert på møte-/endepunkter i nettverket for å ana-

[4]Kan også settes opp med agenter

lysere trafikk fra alle kommuniserende enheter. Applikasjonsbaserte IDS (AIDS) blir ofte brukt som et supplement til NIDS og/eller HIDS for å analysere handlingene som oppstår inne i og mellom applikasjoner. Hvis systemet er implementert riktig kan AIDS monitorere interaksjonen mellom bruker og applikasjon, og ha muligheten til å detektere suspekt aktivitet som for eksempel rettighetseskalering utenfor brukerens tillatte område.

Ulike typer IDS kan også operere på forskjellige måter; enten som signatur- eller anomalibasert. Signaturbaserte systemer analyserer all aktivitet, og leter etter spesielle signaturer som er blitt karakterisert som suspekte. En signatur kan være et sett med handlinger, eller en streng av informasjon i en pakke. Svakheter ved bruk av denne metoden er at alle signaturer må vedlikeholdes i en signaturdatabase, noe som medfører at nye/ukjente angrep ikke detekteres. Det positive er at mengden falske positiver er mindre eller lik null, siden signaturene i databasen allerede er kjente.

Den andre måten et IDS kan operere på er i en såkalt anomal modus. Måten denne fungerer på er at den leter etter avvik fra en "normal" tilstand, og rapporterer om uregelmessigheter dersom denne terskelen overskrides. Denne måten å operere på kan i motsetning til signaturbaserte systemer fange opp nye/ukjente angrep, såfremt den daværende aktiviteten skiller seg fra den normale tilstanden. Ulempen er at anomale systemer kan generere flere falske positiver, da de minste endringer i systemet/nettverket kan fremstå som avvik.

3 Metodikk

Når man tester virkemåten til exploits og/eller annen skadelig programvare, er det svært viktig å bruke en metodikk som er etterrettelig og som både gir pålitelige og nøyaktige resultater. Alle unormale endringer detektert i målsystemet må analyseres og vurderes, og ingenting kan ignoreres. Utfordringen blir her å skille mellom normal og unormal aktivitet.

I dette kapittelet presenteres ulike metoder for denne type kvalitetssikring, samt valg av en eller flere av disse til eksperimentet i kapittel 5.

3.1 Statisk analyse

Statisk analyse (eng: static analysis[14]) kjennetegnes ved at det blir gjennomført på "døde" systemer. Det vil si systemer som ikke befinner seg i en kjørende tilstand. Det kan brukes forskjellige metoder ved statisk analyse, men de mest brukte innen dette området er manuell kodeanalyse og overflateanalyse (se nedenfor).

En av de største ulempene ved statisk analyse er tidsforbruket til metoden dersom systemet er stort eller kildekoden er kompleks.

3.1.1 Manuell kodeanalyse

En mye brukt metode for å undersøke exploits er som nevnt manuell kodeanalyse (eng.: sourcecode review). Dette er en metode hvor man manuelt går gjennom programkoden linje for linje for å avdekke feil, mangler eller forbedringsmuligheter. Ved bruk av denne metoden har man kontroll på hvordan exploiten oppfører seg i systemet, samtidig som systemene rundt også blir ivaretatt for eventuelle ringvirkninger. Poenget med denne metoden er at systemet ikke trenger å kjøres i sanntid mens analysen foregår.

Fordeler	**Ulemper**
Ser hele hendelsesforløpet på kodenivå	Tidkrevende
Ser alle instruksjonene exploiten utfører	Kreves høy kompetanse - god innsikt i programmering
Koden eksekveres ikke, dvs. at det ikke er risiko for påvirkning på systemet og miljøet rundt	Vanskelig å oppdage både semantiske og syntaktiske feil. Man må til enhver tid ha full oversikt over hele kodens struktur
	Vanskelig å lese stor kildekode
	Vanskelig å få tak i kildekode

Tabell 1: Fordeler og ulemper ved manuell kodeanalyse

3.1.2 Overflateanalyse

Den andre statiske metoden er overflateanalyse (eng.: surface-analysis[14]). Dette er en metode som i de fleste tilfeller er betydelig raskere en manuell kodeanalyse, da man kun ser på differansen mellom en gitt/kjent tilstand og en annen ukjent tilstand ved avsluttet test. Før- og etteranalysen skjer på et "dødt" system, dvs. en kopi av tilstanden, eller et montert filsystem. Man vil i hovedsak kun se permanente endringer gjort mellom disse punktene, og man vil raskt kunne avdekke om (og hvor) det har blitt lagt inn rootkits eller annen skadelig kode i målmaskinen. Eventuelle endringer gjort på samme fil *flere* ganger vil derimot ikke oppdages da deres tidligere verdier vil bli overskrevet.

I noen tilfeller vil det også være nødvendig å se mer enn bare sletting, endring og opprettelse av filer. Dette kan for eksempel dreie seg om å undersøke filenes metadata for å se om (og når) de har blitt lest, skrevet eller åpnet. Det at man kan hente ut denne informasjonen (eksempelvis ved å montere filsystemet) gir denne metoden en stort fordel.

Forskjellen mellom overflateanalyse og manuell kodeanalyse er nøyaktigheten: mens overflateanalysen gir opplysninger om filendringer, ser manuell kodeanalyse på enhver instruksjon programmet gjør og gir fullstendig oversikt over hele programmet. Det er derfor stor forskjell i tidsforbruk på disse to metodene, selv om dette avhenger av størrelsen på koden/systemet.

Metodens positive og negative sider blir vist i tabell 2.

Fordeler	**Ulemper**
Teknisk enkelt å utføre	Mister all informasjon som er logget *mellom* start og slutt
Krever lite kunnskap	Får ikke ut all relevant data (startede/stoppede prosesser)
Trenger ikke installert programvare på målmaskin	
Vanskelig for malware å detektere (Mtp. forsvarsmekanismer)	

Tabell 2: Fordeler og ulemper ved overflateanalyse

3.2 Dynamisk analyse

Dynamisk analyse (eng: dynamic analysis[14]) er en teknikk der man følger med på hvordan exploiten oppfører seg under kjøring, istedenfor å finne virkemåte basert på oppbygging. Her logges all aktivitet for ønskede ressurser fortløpende mens operativsystemet kjører, og man har mulighet til å se hvordan exploiten oppfører seg i forhold til ressursene.

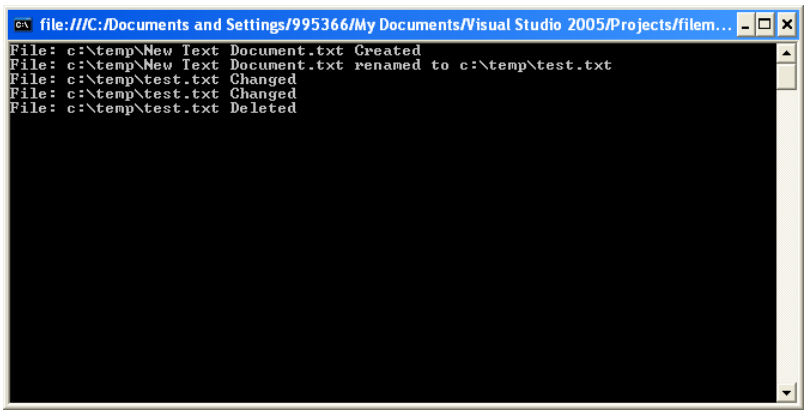
Det finnes flere teknikker for å logge aktiviteten i et operativsystem. En teknikk som ofte benyttes er såkalt "API Hooking"[5][6] hvor man bytter ut en funksjon i operativsystemets API med egen kode. Med denne teknikken kan man avskjære kall til API-funksjoner, kjøre sin egen kode og så gi kontrollen tilbake til operativsystemet.

Hvis man kikker under panseret i et operativsystem, vil man se at all tilgang til filsystemet går gjennom såkalte systemkall. Man kan se på systemkall som et grensesnitt mellom et kjørende program og operativsystemet. Dette grensesnittet er ofte tilgjengelig i form av en rekke assembly-instruksjoner. For blant annet å innføre ekstra sikkerhet opererer de fleste moderne operativsystem med enda et lag over systemkallene. Dette ekstra laget kalles gjerne et system-API. Kort forklart må et program først gå gjennom et programmerings-API og deretter systemkallene før det kan utføres skriving eller lesing av filer.

Nedenfor er en oversikt over noen av de mest brukte teknikkene for API Hooking:

- **File System Filter Driver**
  Windows tilbyr en funksjonalitet som kalles "File System Filter Driver" [15]. Ved å legge inn et filter som avskjærer forespørsler mot filsystemet, har man mulighet til å overvåke og endre filtrafikken. Eksempler på programmer som benytter denne teknikken er antivirusprogrammer, backupagenter og krypteringsløsninger.

- **Overvåke kataloger**
  Noen programmerings-APIer tilbyr funksjonalitet for å overvåke gitte kataloger på filsystemet. Et eksempel på denne teknikken er klassen FileSystemWatcher() i .NET [16]. Ved hjelp av denne klassen kan du angi eventhandlere for endring i filer, når filer blir opprettet, slettet etc. Det er også mulig å legge på et filter hvis du for eksempel kun ønsker å overvåke en gitt filtype. Figur 1 viser et eksempel på bruk av klassen ved overvåking av C:\temp.



Figur 1: Skjermbilde av program som overvåker filendringer i sanntid

[5] Avskjære API (Application Programming Interface)

- **Proxy-bibliotek**
  API-hooking ved hjelp av proxy-bibliotek [17] er en teknikk hvor man bytter ut bibliotekfilene som inneholder API-funksjoner. For å gjøre dette må man først opprette en egen bibliotekfil hvor funksjonsdefinisjonene tilsvarer den i det originale biblioteket. Hvis man da erstatter den originale filen med den nye versjonen, vil operativsystemet begynne å bruke din kode. Hvis ønsket er å logge all filaktivitet vil man typisk legge inn egen kode i funksjoner som read(), write() etc. Ved å gi kontrollen tilbake til det originale biblioteket etter å ha kjørt egen kode, vil det være vanskelig å oppdage at noe uvanlig har skjedd. Denne teknikken går for å være den enkleste måten å hekte seg inn i system-APIet, men vil være upraktisk hvis man skal erstatte bibliotekfiler som inneholder svært mange funksjonskall.

- **Import Address Table Hooking**
  IAT Hooking [18] er en teknikk som utnytter konseptet om dynamisk lasting av delte biblioteker [6]. Når et program startes, hentes det samtidig inn en liste med pekere til alle funksjonene i biblioteket. For å utføre IAT Hooking vil en angriper endre pekeren så den peker på angriperens egen kode. Når dette er gjort kan angriperen overvåke og manipulere alle programmer som bruker biblioteket.

- **Service Dispatch Table Hooking**
  I Windows NT-systemer ligger det et eget lag under system-APIet [7] som utfører den faktiske jobben når noen kaller en funksjon. Dette laget kalles "NT System Services" [19]. For eksempel når en Win32-applikasjon kaller CreateProcess() eller når en POSIX-applikasjon kaller fork(), er det til slutt system servicen NtCreateProcess() som tar seg av eksekveringen. For de som er kjent med UNIX-systemer, kan system services sammenlignes med systemkall i Unix. Den enkleste måten å lage en hook, er å finne tabellen over system services og endre pekeren til å peke på en annen funksjon. Man kan kun endre denne tabellen i kernelmodus, derfor må dette gjøres gjennom en driver. På denne måten kan man overvåke og manipulere alle programmer som benytter seg av system services.

Fordeler	**Ulemper**
Gir mye informasjon	Teknisk komplisert
Informasjon i sanntid	Sannsynligheten for at ondsinnet kode oppdager overvåkingen er til stede
	Krever programvare på målmaskinen

Tabell 3: Fordeler og ulemper ved dynamisk analyse

[6] Dll-er i Windows eller so-filer i Unix
[7] KERNEL32, USER32, etc.

3.3 Binæranalyse

Noen ganger er det nødvendig med en mer grundig analyse av en exploit. For å være 100 % sikker på exploitens virkemåte er man avhengig av kildekoden. Men hva med de gangene man ikke har tilgang til kildekoden? Hvis man ønsker en grundig analyse og man ikke har kildekoden, finnes bare ett godt alternativ: binæranalyse.

Binæranalyse vil si å studere noe i binærformat, dvs. enere og nuller. Når man kompilerer et program så vil det ende opp i dette formatet. Desverre er ikke binærkode spesielt enkel å lese, derfor trenger man verktøy for å gjøre jobben lettere.

En debugger er et program som benyttes for å teste og feilsøke andre programmer. I tillegg til å være et uvurderlig verktøy når man skal finne en feil i et program, har debuggere en annen nyttig funksjonalitet - den kan brukes til "reverse engineering". Kort fortalt er "reverse engineering" en rekke teknikker brukt for å finne ut hvordan et program virker, uten å ha tilgang på kildekoden. En debugger tilbyr blant annet:

- **Disassembler:** Oversette binærkode til assembly-instruksjoner. Noen disassemblere hjelper også til med å holde styr på løkker.
- **Step-by-step:** Gå gjennom programmet instruksjon for instruksjon.
- **Registre og minne:** Mulighet til å se innholdet i prosessorens registre og datamaskinens minne mens man går gjennom programmet.

Fordeler	**Ulemper**
Pålitelig	Tidkrevende
	Kan oppdages av exploiten
	Krever høy kompetanse

Tabell 4: Fordeler og ulemper ved debugging

3.4 Signatur- og modellbasert analyse

Denne metoden brukes for å identifisere eventuell ondsinnet aktivitet basert på tidligere erfaring. Disse erfaringene kan være historiske lagrede data som for eksempel signaturdatabaser samlet over tid (ofte brukt i anti-malware programvare), eller det kan være deteksjon som baserer seg på uregelmessigheter der normal aktivitet sammenlignes mot uvanlig aktivitet og kun avvikene registreres. Dette kan på mange måter sammenlignes med en anomal eller signaturbasert IDS (Se kap 2.5).

For å dra nytte av denne metoden kreves det at man enten skaffer nok data til signaturdatabasen, eller at man lærer målmaskinens normale aktivitet. Førstnevnte kan virke problematisk med tanke på helt nye exploits som enda ikke er registrert/oppdaget,

mens sistnevnte kan by på problemer med tanke på den svært variable oppførselen til et operativsystem. Tabell 5 og 6 viser ulempene og fordelene til henholdsvis signatur- og modellbasert analyse.

Fordeler	**Ulemper**
Få falske positiver	Tidkrevende
	Krever kunnskap om operativsystemet som måles
	Ukjente exploits kan gå udetektert

Tabell 5: Fordeler og ulemper ved signaturbasert analyse

Fordeler	**Ulemper**
Kan detektere ukjente exploits	Stor sjanse for falske positiver
	Krever kunnskap om operativsystemet som måles

Tabell 6: Fordeler og ulemper ved modellbasert analyse

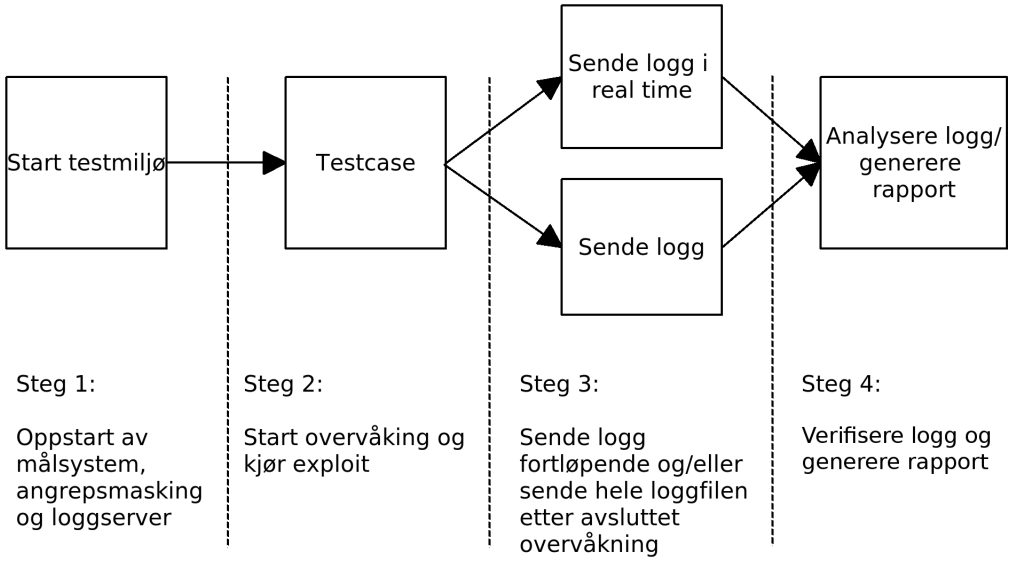
3.5 Automatisering

Automatisering vil si at hele eller deler av en prosess kjører automatisk, med minst mulig menneskelig interaksjon. Automatisering gjør et system enklere å bruke i tillegg til at man øker påliteligheten ved å minske mulighetene for menneskelige feil som kan oppstå ved repeterende oppgaver. Dette kan for eksempel være ved repeterende testing som skal gjennomføres med samme parametere, for å verifisere kodens etterrettelighet.

I et helautomatisert system vil brukeren kun behøve å starte systemet og velge ønsket test for at oppgaven skal bli utført, mens i et delautomatisert system kreves noe interaksjon fra brukeren. Dette kan for eksempel være mellom større operasjoner eller input fra brukeren.

Ved testing og kvalitetssikring er man nødt til å kunne stole på resultatene, og kjøring av samme test flere ganger er nødvendig for å verifisere at man får et konsistent resultat. Ved å automatisere testene har man like kriterier for hver testkjøring, og sannsynligheten for et likt resultat øker.

I figur 2 har vi visualisert de forskjellige fasene som inngår ved kvalitetssikringssystemet.



Figur 2: Automatisering - stegvis

I steg 1 kan det brukes automatisering for å starte systemet og de virtuelle maskinene. Loggserveren må automatisk starte loggsystemet og målmaskinen kan starte å logge aktivitet. Når steg 1 er ferdig er systemet klart for testing. Her kan enten exploiten settes til å starte automatisk, eller så kan brukeren velge og gjøre dette manuelt. Logging kan enten sendes i sanntid, eller når testen er ferdig. Loggserveren analyserer og genererer rapport ut fra gitte kriterier.

Det er mulig å helautomatisere systemet, men dette setter krav til at verktøy og systemer som brukes kan scriptes. Dette vil si at de støtter kommandolinje og ikke krever interaksjon fra brukeren under kjøring. Systemet kan da startes, og all jobben blir utført samtidig som sluttrapporten blir generert.

Oppstart av systemet, loggoverføring, verifisering og rapportgenerering er prosesser som i første omgang er hensiktsmessig å automatisere. Videre er det logisk å bygge videre med prosessene informasjonshenting, prosessering av rådata og sammenstilling av resultater, siden dette er mer krevende jobb.

3.6 Virtualisering

Virtualisering, eller det som gjerne omtales som ”full virtualisering”, er en teknikk der man emulerer en fysisk datamaskin ved hjelp av programvare. I praksis betyr det at man kan kjøre flere operativsystemer oppe på det originale operativsystemet.

Innen sikkerhetsbransjen benyttes ofte virtualisering når man må teste potensiell skadelig programvare, da den skadelige programvaren ikke har mulighet til å nå utenfor det virtuelle miljøet. Dette resulterer i at man unngår å skade de underliggende systemene.

Fordeler	**Ulemper**
Man kan teste programvare i et sikkert miljø	Exploiten kan oppdage at den kjører i et virtuelt miljø og oppføre seg anderledes enn i et ikke-virtuelt miljø

Tabell 7: Fordeler og ulemper ved virtualisering

3.7 Emulering

Emulering er en metode som kan brukes i sammenheng med forskjellige analyser for å kunne overvåke exploits. Ved emulering imiteres det et målsystem som for eksempel kjører programmer tilpasset andre operativsystemer.

De fleste emulatorene deler inn emuleringen i moduler. Modulen emulerer både maskinkoden og operativsystemet slik at de ikke er i kontakt med host systemet.

Fordeler	**Ulemper**
Exploits kan testes i et sikkert miljø	Exploits kan detektere emulering

Tabell 8: Fordeler og ulemper ved emulering

3.8 Fysisk

Et fysisk miljø vil si å bruke fysiske maskiner til testing. Dette krever mer jobb og flere maskiner enn virtualisering, som kan ha flere operativsystemer på én datamaskin.

En stor fordel for testing av ondsinnet kode er at noe ondsinnet kode kan oppføre seg annerledes i et virtuelt miljø i forhold til en fysisk maskin. Dette er for å ikke bli oppdaget i et testmiljø, som ofte er virtuelle. En fysisk testmaskin vil altså, med unntak av eventuelle overvåkningsverktøy som ikke må oppdages, bli mer reell.

Fordeler	**Ulemper**
Exploits vil ikke oppføre seg anderledes	Tungt å bruke/vedlikeholde

Tabell 9: Fordeler og ulemper ved fysisk miljø

3.9 Kvalitetssikring

Ordet kvalitetssikring er definert som ”alle planlagte og systematiske aktiviteter som er iverksatt som en del av kvalitetssystemet og påvist som nødvendig for å skaffe tilstrekkelig tiltro til at en enhet vil oppfylle kravene til kvalitet.”[20]. Tiltro er et viktig begrep i denne sammenhengen. Systemet må kunne teste exploits på en slik måte at det skapes en tiltro til at aktivitet som utføres av exploiten faktisk blir registrert.

For å kunne kvalitetssikre exploitene på en fornuftig og vitenskapelig måte, bør det settes ned kriterier som exploitene skal testes mot. Hvis exploiten ikke kan tilfredstille en av disse kriteriene, kan ikke denne godkjennes. Under vises en liste med kriteriene vi har satt opp med en påfølgende forklaring.

Eksempler på kriterier for å kvalitetssikre exploits:

- Ondsinnede bivirkninger (exploit med malware)
- Andre bivirkninger
- Krav til repeterbarhet / samme resultat flere ganger
- Systemet er egnet til automatisering

3.9.1 Ondsinnede bivirkninger

Exploits i dag finnes i mange forskjellige varianter og er definert som kode som utnytter sårbarheter i programvare. Hvis en angriper utnytter en sårbarhet, kan han legge til sin egen payload[8] for å få målsystemet til å oppføre seg annerledes. Ved å legge til malware i exploit eller payload, kan dette føre til ondsinnede bivirkninger hos både målsystem og angrepsmaskin. Dette kan være bivirkninger som å slette viktige filer/prosesser, sende sensitiv informasjon ut på internett eller laste ned mer malware. Hvis exploiten som testes inneholder for mange negative bivirkninger, bør ikke denne brukes i sammenheng med penetrasjonsstesting da dette kan skape store problemer for både stabiliteten og sikkerheten i systemet.

3.9.2 Andre bivirkninger

Andre bivirkninger som kan påvirke exploiten er at den i seg selv er skrevet for dårlig eller inneholder feil. Exploiten kan gjøre ondsinnende handlinger som beskrevet over, gjøre noe annet enn det som var hensikten til programmereren, eller den kan oppføre seg forskjellig fra gang til gang. Slike exploits bør man være forsiktige med å bruke, selv om baktanken med exploiten kan være positiv.

[8] Kode som eksekveres i det en sårbarhet er utnyttet

3.9.3 Repeterbarhet

For at en exploit skal kunne kvalitetssikres er det viktig at den oppfører seg identisk hver gang. Hvis den oppfører seg forskjellig, blir det vanskelig å verifisere om den kan være aktuell i penetrasjonsstesting. Ved ulike testresultater bør resultatene kombineres for å finne all aktivitet. Repeterbarhet kan derfor spare mye tid og man blir tryggere på endringene til exploiten.

3.10 Oppsummering

De forskjellige metodikkene har alle sine fordeler og ulemper ved analyse av exploits. Vår løsning må kunne implementeres på flere plattformer, samtidig som den er enkel å bruke og oppfyller kravene til oppgaven. Det er som nevnt tidligere i rapporten ønskelig at systemet skal være hel- eller delautomatisert, noe som også vil være med å påvirke valg av metodikk.

Statisk analyse er en metode som krever lesing av kildekode, og er noe vi i utgangspunktet vil bevege oss vekk fra. Binæranalyse krever også innsikt på maskinkodenivå, men da ved hjelp av et debugger-program som viser instruksjoner og hvordan kodeforløpet blir. Dette mener vi at også blir for manuelt til bruk i praksis, og står da ikke i stil med ønsket om automatisering. Signatur- eller modell-basert analyse, som baserer seg på historikk og mønster vil kreve at det hele tiden er nok historikk/kunnskap til alltid å kunne detektere de nyeste truslene. Siden nye exploits kan ha nye teknikker for å skjule evt. ondsinnede aktivitet, vil det være svært vanskelig å holde seg oppdatert til enhver tid.

Dynamisk analyse kan brukes til å logge endringer som skjer når et operativsystem kjører. Ved analysering av rådata vil vi kunne se de aller fleste endringer som er gjort, og på denne måten verifisere om koden kan kvalitetssikres eller ikke. Overflateanalyse kan se på forskjellen mellom to forskjellige tilstander til en maskin. Vi kan for eksempel se på filsystemet før og etter kjøring av exploiten og observere hvilke filer som har blitt opprettet, endret, slettet eller flyttet.

Ved å kombinere dynamisk analyse og overflateanalyse, ser vi for oss at det vil forekomme et tilfredsstillende resultat. Overflateanalyse vil hjelpe oss å finne endringer i filer, register og minne, noe som blant annet kan hjelpe oss og finne ut om noe er lagt til i systemet for senere kjøring eller som en bakgrunnsprosess/rootkit. Dynamisk analyse vil logge endringer som skjer når systemet kjører. Dette gjør at vi ikke mister informasjon om hva som skjer når exploiten kjører. For eksempel kan vi se om det blir sendt ut informasjon over internett. Testmetodikken vil dermed bevege seg vekk fra white-box tankegang og fokusere mer på grey-box testing hvor vi har noe kjennskap til både miljø og kode på forhånd. Dette vil gi en løsning som gir oversikt over endringer og handlinger som skjer i systemet på en hel- eller delautomatisk måte. På grunnlag av tilgjengelig utstyr, og det å ha mulighet for å tilbakestille testmiljøene, har vi valgt å basere oppgaven på en virtuell løsning. I utgangspunktet har det også vært ytret et ønske om dette fra ekstern veileder.

4 Konseptuelt rammeverk og testverktøy

I dette kapittelet vil det på et konseptuelt nivå fremstilles et rammeverk for å kvalitetssikre exploits. Rammeverket bygger på eksisterende resultater innen malware-analyse og intrusion detection. I de første seksjonene vil det fokuseres på datainnsamling, hvor de to metodene overflate og dynamisk analyse blir presentert. Videre vil det fremkomme analyse av data, og til slutt presentasjon av resultatet. I tillegg vil det konsekvent gjennom hele kapittelet fokuseres på ulike typer verktøy som kan brukes i sammenheng med datainnsamlingen. Løsninger rundt testmiljø samt metoder tatt i bruk for innsamling, analysering og presentasjon av data vil også omtales i kapittelet.

4.1 Fremgangsmåte

Fremgangsmåten som figur 3 viser, består av de samme fasene som beskrevet i kapittel 2.4 (Digital etterforskning): innsamling, analyse og presentasjon. Prinsippene er de samme, men metodene og fremgangsmåten kan variere.

Figur 3: Overordnet fremgangsmåte

1. **Datainnsamling**
   Datainnsamlingen er det første trinnet i prosessen. Det er i denne fasen at alle rådata samles inn for videre analyse, og hentes ofte inn fra flere kilder.

2. **Dataanalyse**
   Etter at alle rådata er samlet inn, må de analyseres/filtreres for å få ut de mest relevante dataene. Ettersom det er samlet inn data fra flere kilder, blir det også nødvendig å sammenligne disse. Hvilke data som beholdes eller fjernes kommer an på en rekke forhåndsdefinerte regler som vi kommer nærmere inn på senere.

3. **Presentasjon**
   Denne delen av analysen står for hvordan data skal presenteres for brukeren. Selv om analysen henter ut de mest relevante dataene, er de fortsatt i et format som ikke alltid er like lettlest. Viktige momenter i presentasjonen er blant annet hvilket detaljnivå brukeren skal få, utheve viktige data og fremstille data ved hjelp av grafer og illustrasjoner.

4.2 Datainnsamling

For å kunne overvåke hva en exploit gjør med målmaskinen er det viktig å se hva som endrer seg av ressurser når en exploit blir kjørt. Ut ifra oppgaven har vi delt ressursene inn i filer, prosesser, nettverk, register og minne. Det er nødvendig å inkludere alle ressursene for og få et komplett resultat, og eventuelt kunne godkjenne eller underkjenne en exploit.

Første steg ved analysering av en exploits er innsamling av all data/aktivitet på målsystemet. Da det er denne rådataen som bestemmer både detaljnivået og testens kvalitet blir dette selve grunnlaget for videre analyse og presentasjon.

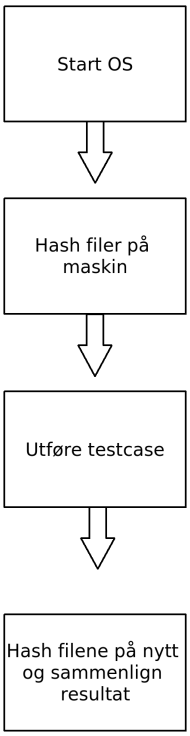
Som nevnt tidligere så skal sluttresultatet presenteres på tre forskjellige nivåer. Det første resultatet skal gi et sammendrag av testen og gi en indikasjon på om testen passerte kvalitetssikringen eller ikke. Det andre skal gi mer utfyllende detaljer, og det siste resultatet skal presentere rådataene til bruk for videre analyse. Det er derfor nødvendig å tilby en tilstrekkelig mengde med data, samtidig som det må være så relevant som mulig.

I de neste seksjonene vil det vises hvordan overflateanalyse og dynamisk analyse blir utført, deretter presenteres det metoder for hvordan datainnsamling kan foregå med disse metodene for filer, prosesser, register, nettverk og minne.

4.2.1 Overflateanalyse

Overflateanalyse er som tidligere nevnt en underliggende metode av statisk analyse. Statisk analyse vil si å analysere systemer i fryst tilstand, som for eksempel montering av et ”dødt” filsystem ved hjelp av vmware-mount.pl[21] eller manuell kodeanalyse. Overflateanalyse er ofte noe av det første man foretar seg når man skal undersøke endringer mellom to tilstander. Ved å kun se på differansen mellom start- og sluttilstanden til systemet som overvåkes, får man ofte en grov indikasjon på hva exploiten har til hensikt og gjøre med målsystemet.

I og med at man kun ser de permanente endringene, vil ikke analysen gi noe informasjon om hva som skjedd mellom start- og sluttilstanden. Da dette kan være informasjon av interesse, vil det heller bli supplert med mer dynamiske tilnærminger for å fylle dette hullet. Videre er overflateanalysen rask til å avdekke endringer gjort av exploiten, samtidig som den er sikker med tanke på at det jobbes på et dødt system, og da ikke har mulighet for å påvirke systemet eller de innsamlede dataene. Figur 4.2.1 viser overflateanalysens fremgangsmåte.



Det første steget i datainnsamlingen er å starte operativsystemet. Her må man sørge for at operativsystemet har gjort ferdig sine oppstartsrutiner for at alle tester skal bli gjennomført fra et likt utgangspunkt.

Når maskinen er startet må man ta et fingeravtrykk av alle filene på maskinen. Dette for å kunne ha et utgangspunkt og sammenligne med når testen er gjennomført. For å få med alle endringer som blir utført av exploiten, lages det fingeravtrykk av hele filstrukturen.

I dette steget utføres forskjellige tester i testmiljøet. Alle endringer som påvirker de forskjellige ressursene blir logget i påvente av videre analyse.

Filsystemet hashes på nytt når exploiten har blitt eksekvert. Deretter sammenlignes de to hashene for å detektere hvilke endringer som har blitt påført målsystemet. Videre kan det suppleres med innsamling av filenes metadata for å hente ut mer om handlingsforløpet i systemet.

Start OS
Når datainnsamlingen startes, ønsker vi at OSet er ferdig med alle oppstartsrutiner for å unngå all unødvendig aktivitet. Når man starter OSet på den ordinære måten, blir det derfor nødvendig å vente til alle oppstartsrutiner er ferdig, eller utvikle en teknikk for og stadfeste at OSet er klart til bruk. Det andre alternativet, er at man vekker OS-et fra en dvaletilstand, som gjør at det er mulig å unngå disse problemene ved at OSet blir klargjort før testen. Så når testen starter, kopieres det inn et klargjort speilbilde som kan vekkes til live.

Ettersom testmiljøet baserer seg på virtualisering, har vi to alternativer:

- Starte OSet på vanlig måte
- Vekke OS fra dvaletilstand (eng. suspended state)

Start hashing av filer på maskin
Ved overflateanalyse jobbes det på på et "dødt" målsystem. For å kunne ha et utgangspunkt og sammenligne med, må hele målsystemet dokumenteres. Dette gjøre som oftes ved å ta et "fingeravtrykk" av hele systemet, også kalt hashing/fingerprinting. For dette finnes det forskjellige algoritmer som kan brukes, og som har forskjellige egenskaper slik at de kan skreddersys til ulike bruksområder. For å kunne spore alle endringer som blir påført målsystemet, må hele filstrukturen hashes. Dette trengs kun å gjøres en gang, da man ved senere tester benytter samme utgangspunkt.

Utføre et testcase
I denne fasen er testmiljøet klart til bruk og selve testen skal utføres. Ut ifra hvilken type exploit som skal testes, vil utførelsen være noe forskjellig. Man kan dele opp exploitene i to kategorier basert på hvordan de eksekveres:

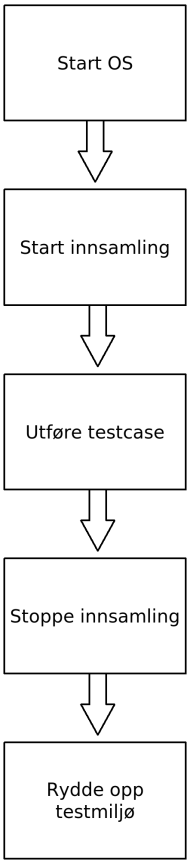
- **Lokal exploit**
  Dette er exploits som skal eksekveres lokalt på datamaskinen. Dette kan være alt fra en exe-fil som utnytter en feil i et bibliotek, til en buffer overflow i en pdf eller mp3-fil. Felles for disse er at man er inne på samme maskin som overvåknings-agenten for å utføre testen. Et problem med dette er at det vil kreve litt navigering rundt i operativsystemet, noe som vil genereres ekstra trafikk. En mulig løsning vil være å eksekvere exploiten ved hjelp av RPC (Remote Procedure Call) eller vmrun[9], og på denne måten kjøre exploiten fra en ekstern maskin.
- **Exploit over nettverket / Remote exploit**
  Dette er exploits som ikke eksekveres lokalt på målsystemet, men i stedet utføres over nettverket. Exploitene kan eksekveres fra en annen maskin i nettverket, eller bli lastet ned fra internett. Et eksempel på dette er en exploit som utnytter en nettverkstjeneste som for eksempel Microsoft IIS med en buffer overflow.

Hashing av filer og sammenligning
Når testcasen er gjennomført må de permanente endringene som exploiten har påført systemet lokaliseres. Ved å hashe hele filstrukturen på nytt og sammenligne dette med den hashdatabasen som ble gjennomført før testcasen, kan det enkelt hentes ut hvilke filer som er endret. Videre kan det bli benyttet ulike teknikker for å se etter endringer i filenes metadata for å stadfeste handlingsforløpet til de enkelte filene.

[9]Kommando for å styre virtuelle maskiner i VMware

4.2.2 Dynamisk analyse

Ifølge whatis.com kan man definere dynamisk analyse som: "..the testing and evaluation of a program by executing data in real-time"[22]. Stikkordet her er "real-time". All data samles inn i sanntid ettersom endringene skjer på operativsystemet. Selv om dynamisk analyse tilbyr en rekke fordeler, er det også mer komplisert enn andre teknikker som overflateanalyse. Den største utfordringen er å få tak i dataene, men det finnes en rekke teknikker man kan benytte.



Det første steget i den dynamiske analysen er å starte operativsystemet. Her må man sørge for at operativsystem har gjort ferdig sine oppstartsrutiner for at alle tester skal bli gjennomført fra et likt utgangspunkt.

Ved dynamisk analyse logges data som genereres mens systemet kjøres. For å samle inn disse dataene, benyttes en agent som logger alle endringer i sanntid.

I dette steget utføres forskjellige tester i testmiljøet. Alle endringer som påvirker de forskjellige ressursene blir logget i påvente videre analyse.

Etter en gitt tid, når man er sikker på at testcaset er ferdig, stoppes agenten som overvåker systemet. Man sitter da igjen med rådata som må sendes videre til analyse.

Når innsamlingen er gjennomført, må man overføre rådata til maskinen som skal utføre analysen. Det siste som skjer er at testmiljøet blir resatt og klargjort for ny test.

Overflateanalyse og dynamisk analyse har flere likheter i oppsettet. Det er derfor i avsnittene nedenfor kun vist de to metodenes ulikheter.

Start innsamling
Ved dynamisk analyse logges data som genereres mens systemet kjøres. For å samle inn disse dataene, benyttes en agent. Kort fortalt er agenten et program som kjører på systemet som skal analyseres. Det er også mulig å hente ut dynamiske data fra utsiden av OSet[23], men dette er en teknisk komplisert løsning. En enklere løsning er å basere seg på agenter. Selv om en agent gir fordeler ved at vi får en mindre teknisk komplisert løsning, har den også noen ulemper. Den største fordelen er også den største ulempen

- agenten kjøres på det samme systemet hvor testen skal utføres. Først og fremst vil agenten kunne påvirke testen. Et eksempel er overvåking av filer: hvis agenten skriver til fil, vil det vises i loggene. Den andre store ulempen ved å ha en agent på testsystemet er at testsubjektet, i dette tilfellet en exploit, vil ha mulighet til å oppdage og i verste fall påvirke agenten. Hvis exploiten stopper agenten, eller modifiserer noen av dataene som blir logget, vil vi ikke kunne stole på testen.

Stoppe innsamling
Et viktig spørsmål her er hvor lenge man skal vente før man stopper innsamlingen. Skal man stoppe så fort exploiten har eksekvert, eller skal man vente en gitt tidsperiode? Poenget her er at man må være sikker på at exploiten har gjort seg ferdig, før man stopper innsamlingen. En enkel buffer overflow vil eksempelvis være ferdig i samme øyeblikk som exploiten er ferdig med å kjøre, men hvis den i tillegg legger inn malware kan det komme frem flere detaljer over tid. Dette kan for eksempel være en logisk bombe.

Rydde opp testmiljø
For at alle tester skal ha et likt utgangspunkt, er det viktig at testmiljøet blir satt tilbake til den tidligere kjente tilstanden. Dette kan gjøres på forskjellige måter, blant annet ved speiling av fysisk harddisk eller snapshot ved virtualisering.

4.2.3 Datainnsamling - Filer

Filer er en vesentlig ressurs å ta hensyn til når man skal etterforske hvilke endringer som er gjort på et målsystem. Hvis man for eksempel ser på oppbyggingen til UNIX, er absolutt alt basert på filer. Det er riktignok tre forskjellige kategorier; rene filer, mapper og spesial- eller enhetsfiler (CD-rom, printer, USB-enheter osv.), men til syvende og sist ender alt opp med lesing/skriving til fil. Det er derfor nærliggende å se på filsystemet og hvilken informasjon det inneholder for og kartlegge maskinens aktivitet. Filsystemet kan blant annet gi oss informasjon om filenes metadata og såkalte MAC-times, som blant annet lar oss se når filer er blitt lest, åpnet eller endret.

Overflateanalyse
Det finnes flere teknikker for å detektere endringer i filer, men innen statisk analyse er de mest brukte hashing og monitorering av tidsstempler (ref. metadata og MAC-times). I hvert av tilfellene kan det utføres en sammenligning av en tidligere kjent tilstand mot en ny ukjent tilstand, og forskjellene avslører hvor endringene har forekommet. Den førstnevnte teknikken, hashing, er en primitiv og pålitelig teknikk der det lages et unikt fingeravtrykk av hver fil før og etter utført test, for så og sammenligne disse fingeravtrykkene etterpå. Der hvor fingeravtrykkene er endret er også filens *innhold* endret. Endring av filens metadata detekteres derimot ikke. Eksempel på metadata til filer kan være: navn, bruker- og gruppetilhørighet, rettigheter og størrelse.

En anerkjent og pålitelig metode for å lage disse fingeravtrykkene er MD5 (Message-Digest algorithm 5) [24]. MD5 tar strenger av data som input, og produserer en unik 128-bit fingerprint av dette. Selv om det er funnet en svakhet (angrepsmulighet) med

kollisjoner av hash-verdier[25], anses ikke dette som en stor risiko for formålet i dette prosjektet, da en dynamisk analyse av samme målmaskin vil avsløre eventuelle avvik fra den statiske analysen.

Det positive ved bruk av MD5 fremfor andre algoritmer er at den ofte benyttes i anerkjente verktøy innen dataetterforskning. Eksempler på dette kan være Helix[26], EnCase[27] og FTK[28]. På bakgrunn av dette kan resultatene fremstilt i testmiljøet sees på som kompatible opp mot andre metoder/resultater.

Eksempler på verktøy som forenkler bruken av MD5 er *MD5sum* for UNIX og *MD5sums* for Windows. Da begge disse verktøyene er kommandolinjebaserte gjør det jobben med automatisering og videre scripting svært enkel. MD5sums i Windows har også muligheter for å sende output i UNIX-format slik at resultater kan sammenlignes med tilsvarende verktøy i UNIX. I noen tilfeller ønsker man kanskje å ha muligheten til og utføre analysen på en annen plattform enn den plattformen målsystemet kjører. At verktøyene som brukes er kompatible mellom flere plattformer er da en stor fordel.

Et eksempel på bruk av MD5sums for å lage fingeravtrykk av hele filsystemet i Linux kan være ”find / -type f | xargs -d ”\n” md5sum”. Dette traverserer først hele filsystemet for å finne filer, og deretter sendes dette videre til MD5sum som genererer fingeravtrykket.

Selv om det er vesentlig å se hvilke filer som har endret seg, er ikke dette alltid nok for å kunne stadfeste det fullstendige handlingsforløpet. Det har seg slik at ved å differensiere fingeravtrykkene til en fil fra to forskjellige tilstander, ser man kun om *innholdet* i filen er endret. Det er derfor ofte ønskelig å gå litt lenger, blant annet for og finne ut om en fil er blitt åpnet, endret eller skrevet til. En måte man kan detektere nettopp dette på, er å se på informasjon fra *MAC-tidspunktene* fra filene som overvåkes. MAC står for **M**odified **A**ccessed **C**hanged, og viser tidspunkt for endring på nettopp disse områdene. Atime (Accessed) oppdateres når en fil er blitt åpnet/lest, eller i det man lister en mappes innhold. Mtime (Modified) viser når filens innhold ble endret, og Ctime (Change) viser når innholdet av metadata i filen ble endret. Det som er bra med denne teknikken er at MAC-tidspunkt finnes på alle filer på alle plattformer, selv om det kan tolkes annerledes fra OS til OS[10]. Siden MAC-tidspunktene er statisk informasjon som er knyttet til enhver fil, er det derfor hensiktsmessig å utføre en statisk analyse (herunder overflateanalyse) for å trekke ut denne dataen uten å påvirke filsystemet. Ved å montere filsystemet som skal analyseres på et annet filsystem (eksempelvis vha. vmware-mount.pl [21]), har man full kontroll over filsystemet, og flere forskjellige verktøy kan deretter benyttes for videre analyse. Det faktum at vmware-mount monterer filsystemet i en read-only tilstand, gjør at det ikke er mulig å påvirke filenes Atime-tidspukt i det man gjennomgår filene eller bare traverserer filsystemet. Nedenfor vises et enkelt script skrevet av Harlan Carve for å finne MAC-tidspunktene.

[10] I Windows tolkes C-time som CreationTime (tidspunkt for opprettelse) og ChangeTime (tidspunkt for endring av innhold)

Listing 1: Script laget i Perl for å trekke ut MAC-tidspunkt på filer

```
use strict;
my $file = shift;

my ($size,$atime,$mtime,$ctime) = (stat($file))[7..10];
my $a_time = localtime($atime);
my $m_time = localtime($mtime);
my $c_time = localtime($ctime);
print "Modification Time:  $m_time\n";
print "Last Access Time:  $a_time\n";
print "Creation Time:  $c_time\n";
```

Det er verdt å merke seg at henting av MAC-times kun trengs å gjøres én gang ved gjennomført test i motsetning til hashing, som må kjøres både før og etter. Dette fordi MAC-tidspunktene fortløpende blir overskrevet, og poenget med sammenligning blir da borte.

Kjente problemområder innen statisk analyse

1. **Overflateanalyse på et filsystem som ikke er read-only**
   Scenario: Systemet som exploiten skal testes på er klargjort (laget database av fingeravtykk til alle filer), og testen startes. Ved endt test vil man da kjøre en ny runde med fingeravtrykk for å finne ut hva som er blitt endret. En ny Hash-database blir laget, og sammenlignes med den fra den kjente tilstanden. Til slutt setter man i gang med uthenting av MAC-tidsstempler.

   Det som her vil skje er at alle Atime stemplene blir satt til samme tidspunkt som fingeravtrykkene tok sted. Grunnen til dette er så enkel som at ved å lage et fingeravtrykk (eksempelvis vha. MD5) så må filens innhold leses, og Atime stempelet blir overskrevet. Dette problemet kan enkelt bli løst ved å først utføre MAC-time innhenting *før* fingeravtrykkene blir laget, men den beste måten å unngå problemet på er fortsatt og montere filsystemet som read-only.

2. **MAC: Filer som blir aksessert/skrevet til gjentatte ganger**
   Scenario: Målsystemet er klargjort, og exploit er eksekvert. Gjennom testperioden blir filene lest, skrevet og endret gjentatte ganger.

   Det å skaffe seg et nøyaktig bilde av hva som har skjedd, samt sekvens/rekkefølgen endringene har blitt gjort kan vise seg å være vanskelig kun ved bruk av MAC-tidspunktene. Hver gang en fil blir modifisert, lest eller endret blir det forrige tidsstemplet overskrevet av det nye.

   Det samme problemet kan oppstå hvis en fil eksempelvis blir injisert med ondsinnet kode, den samme filen eksekverer koden og til slutt gjenopprettes til dens originale tilstand. Det er da i etterkant svært vanskelig å stadfeste (ut i fra MAC-tidspunktene) når den aktuelle filen ble injisert eller og knytte denne filen opp mot handlingene som ble utført.

3. **Obfuskering av tidsstempler**
   En annen ulempe ved bruk av MAC-tidsstempler er at disse kan forfalskes/obfuskeres ved bruk av relativt enkle funksjoner som *utime()* i UNIX og SetFileTime() i Windows. For å ytterligere "forenkle" dette er det blitt laget verktøy som for eksempel "timestomp"[29] og "setFileDate"[30]. Deaktivering av Atime-lagring i et filsystem er også mulig i både Windows og UNIX. I Windows kan registernøkkelen "*HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate*" endres til 1 for og deaktivere Atime fullstendig. I UNIX systemer kommer man unna Atime ved å montere filsystemet med parameteren *"noatime"*.

   Listing 2: Eksempel på endring av MAC-tidspunkt ved bruk av "Timestomp"

```
timestomp.exe c:\test.txt -z "Saturday 10/08/2005 2:02:02 PM"
timestomp.exe c:\test.txt -a "Saturday 10/08/2005 2:02:02 PM"
```

4. **Kode som kun jobber i minnet**
   Det å overvåke filendringer kan være en omfattende jobb i seg selv, og det hjelper ikke at det finnes flere metoder tilgjengelig for og eksplisitt unngå fil-aktivitet på målsystemet. Penetrasjonstestingverktøy som Canavas[31] og Core Impact[32] har benyttet slike teknikker i en lengre periode for å skjule sine spor, men også opensource-rammeverket, Metasploit[7], har i senere tid implementert og tilgjengeliggjort denne teknikken. Da henholdsvis vha. payloaden "meterpreter". Dette bidrar til å vanskeliggjøre kvalitetssikringen av exploits med tanke på en eventuell ondsinnet payload som aldri viser aktivitet mot filsystemet.

   For å gjennomføre et angrep av denne typen så kreves det at det finnes et program på målmaskinen som oppfører seg som en tjener og som kommuniserer med operativsystemet. Denne tjeneren kan enten være en Inter Userland Device (IUD) som gir tilgang til dens egne adresseområde, eller det kan være en IUD som gir tilgang til et annet adresseområde. Teknikkene som ofte brukes i disse sammenhengene er Remote Library Injection[33] og Syscall Proxying[34].

Verktøy for overflateanalyse

Det finnes flere verktøy laget spesielt for innhenting av både fingeravtrykk og MAC-tidspunkt. To av de mest kjente innen forrensics området er Mac-robber[35] og mactimes[36]. Mac-robber er et etterforskningsverktøy basert på det tidligere utviklede verktøyet Grave-robber[37] i TCT (The Coroner's Toolkit) av Dan Farmer og Venema Weiste. Forskjellene er at grave-robber er utviklet i Perl, mens Mac-robber er utviklet i C og er mer spesifikt mot endringer av MAC-tidspunkt. Verktøyet samler inn MAC-dataen fra valgte filer og sender resultatet til enten skjerm eller fil. Videre tar Mactime seg av analysering av rådataene, og presenterer filaktivitet i en tidslinje som er lett og følge.

Listing 3: Eksempel på bruk og output fra Mac-robber og mactime

```
./mac-robber /mnt/VirtualOS/test > data_log.mac && mactime -b data_log.
    mac
...
Tue Feb 17 2009 12:31:40 0 ..c -rw------- 0 0 9801 /mnt/VirtualOS/test/
    fil1.txt
Tue Feb 17 2009 12:31:57 0 ma. -rw------- 0 0 9802 /mnt/VirtualOS/test/
    fil2.txt
Tue Feb 17 2009 12:32:02 0 m.c drwx------ 0 0 9774 /mnt/VirtualOS/test/
    fil3.txt
....
```

Det finnes også verktøy tilgjengelig for å kombinere og automatisere både fingeravtrykk og MAC-tidspunkt. Ett av de beste innen dette, og som også har åpen kildekode, er Afick[38]. Dette er en rask open-source integritetsmonitor designet for å kjøre på alle plattformer som har Perl og dens standard moduler. For de som tidligere har brukt tripwire[39], så vil det gjenkjennes at Afick er veldig likt syntaksmessig.

Det som skiller Afick fra andre programmer er at den har mulighet for å detektere endringer i mer en bare hash-signaturer, fra for eksempel MD5 og SHA, men også tar hensyn til metadata som MAC-tidspunkt, inode, rettigheter, størrelse, antall linker, bruker- og gruppetilhørighet. En ulempe er at programmet ikke er stand-alone, men i stedet er avhengig av en konfigurasjonsfil som må justeres etter behov. Resultatet viser i detalj alle filer som er endret, slettet og opprettet, sammen med en oversiktlig oppsummering (listing 4).

Listing 4: Eksempel på et sammendrag fra Afick

```
...
# Hash database : 21 files scanned, 6 changed (new : 2; delete : 2;
    changed : 2;
dangling : 0; exclude_suffix : 1; exclude_prefix : 0; exclude_re : 0;
    degraded : 0)
# ###################################################################
# MD5 hash of /var/lib/afick/afick => Ia4pM91NbVDRXhLFxpQwwA
# user time : 0.24; system time : 0.02; real time : 0
# ###################################################################
...
```

Felles for alle metoder og verktøy til bruk i overflateanalyse er at de burde utføres på et read-only filsystem. Dette er allerede en kjent arbeidsmåte innen digital-forensics, der ”Order of Volatility” og ”Chain of Custody”-prinsippene står sentralt. Poenget er at resultatet ikke kan bli endret og må forholdes så troverdig som mulig.

Dynamisk analyse

For å dekke problemene nevnt med overflateanalyse, blant annet skriving/aksessering flere ganger eller obfuskering av tidsstempler, kan vi benytte oss av dynamisk analyse. Dynamisk analyse logger endringer som skjer i sanntid, under kjøring av operativsystemet. Endringer som at en fil blir lest eller modifisert flere ganger, vil dermed registreres. For å få til denne loggingen kan det, som nevnt tidligere, benyttes såkalt API-hooking.

API-hooking vil si å koble seg mellom operativsystemet og systemkallene til operativsystemet. Et systemkall kan for eksempel være at man ber operativsystemet om å åpne en fil. På denne måten logges alt som skjer under kjøring, og vi kan videre analysere dette etter at vi er ferdig med å kjøre en exploit.

Som nevnt i kapittel 3 (Metodikk), finnes det mange måter å hente ut dynamiske data fra et operativsystem. For å hjelpe oss med dette kan man blant annet benytte et verktøy fra Sysinternals som heter Process Monitor. Dette er et verktøy som overvåker all aktivitet i filer, prosesser og Windows register. Process Monitor er egentlig et verktøy som er laget for å feilsøke, men det har mulighet til og bli scriptet, noe som gjør at det egner seg som en agent. Dataene lagres først og fremst som et internformat, men kan lett konverteres til CSV.

Når man overvåker filsystemet er vi interessert i å vite om filer har blitt opprettet, skrevet, lest eller slettet. Nedenfor er det listet filoperasjonene, samt eksempeldata fra Process Monitor.

- **Om en fil er opprettet**
  Her kan man se at prosess ”cmd.exe” har opprettet filen C:\temp\dyntest.txt

```
"10:59:48,7587652","cmd.exe","2584","CreateFile","C:\temp\
    dyntest.txt","SUCCESS","Desired Access: Generic Write,
    Read Attributes, Disposition: OverwriteIf, Options:
    Synchronous IO Non-Alert, Non-Directory File, Attributes:
     N, ShareMode: Read, AllocationSize: 0, OpenResult:
    Created"
```

- **Om en fil er skrevet**
  Her har prosessen ”cmd.exe” skrevet til filen C:\temp\dyntest.txt. Man kan se at det ble skrevet 30 byte, og at man begynte på byte 0 i filen. For hver skriveoperasjon blir det registrert to linjer med identisk filnavn i loggen. Den andre linjen viser at det ble utført en skriveoperasjon av prosessen ”System” på samme målfil som ovenfor. Hvis man ser på ”Offset” og ”Length”, vil man se at skrivingen begynte på samme posisjon som ovenfor, men istedenfor å ha en lengde på 30 byte, blir det skrevet 4096 byte. Dette er samme størrelse som filsystemets diskblokker, så det virker som den første operasjoen kun var skriving til cache, mens den andre operasjonen sørget for at cachen ble skrevet til harddisken.

```
"10:59:48,7677678","cmd.exe","2584","WriteFile","C:\temp\
    dyntest.txt","SUCCESS","Offset: 0, Length: 30"

"10:59:49,6823860","System","4","WriteFile","C:\temp\dyntest
    .txt","SUCCESS","Offset: 0, Length: 4 096, I/O Flags: Non
    -cached, Paging I/O, Synchronous Paging I/O"
```

- **Om en fil er lest**
  Her har prosessen "cmd.exe" lest fra filen C:\temp\dyntest.txt. Det ble lest 30 byte fra posisjon (byte) 0.

```
"11:00:06,1765024","cmd.exe","2584","ReadFile","C:\temp\
    dyntest.txt","SUCCESS","Offset: 0, Length: 30"
```

- **Om en fil er slettet**
  Her har prosessen "cmd.exe" markert filen C:\temp\dyntest.txt for sletting. "SetDispositionInformationFile" er en generell operasjon, så man må også se etter om flagget "Delete: True" er satt.

```
"11:00:18,6530733","cmd.exe","2584","
    SetDispositionInformationFile","C:\temp\dyntest.txt","
    SUCCESS","Delete: True"
```

4.2.4 Datainnsamling - Prosesser

Prosesser og tråder er den andre ressursen vi skal se på. Det er vesentlig å overvåke prosesser og tråder for å kvalitetssikre en exploit siden denne ressursen utfører handlinger og kall til operativsystemet.

Prosesser og tråder kjører ikke når operativsystemet er stoppet, men det er likevel teknikker man kan benytte for å hente ut prosessinformasjon fra et dødt system. En mulighet er å hente ut prosesslister fra en minnedump. Man kan for eksempel benytte PTfinder[40]. Siden overflateanalyse og uthenting av prosesslisten må gjøres fra minnet, dekkes dette av minneanalyse-kapittelet. En annen teknikk er å utnytte det faktum at Windows forhåndshenter (Eng.: Prefetch) eksekverbare filer. Dette gjør at man kan se hvilke filer som er kjørt ved å se hvilke filer som er opprettet i katalogen "C:\WINDOWS\Prefetch". Den dynamiske analysen gir mer detaljert informasjon om prosessene, men dette er teknikker som kan være nyttig som et supplement.

Dynamisk analyse kan brukes på to måter. Den første metoden er ved å sammenligne prosesser som kjører før exploiten starter, for så og sammenligne prosessene etter at exploiten har kjørt ferdig. Dette blir en tilnærming til overflateanalyse, men går under dynamisk analyse siden systemet ikke er dødt. Den andre metoden er å overvåke prosessene dynamisk, under kjøring av exploiten. Det vil si å logge mest mulig nødvendig informasjon om prosesser og tråder. Det er viktig å samle informasjon om prosesser og

tråder som starter eller blir avsluttet, og hvilke prosesser og brukere som utfører handlingene. Informasjon som også er ønskelig å logge er blant annet "PID"[11], prosessnavn, full bane til den eksekverte filen og eventuelle delte biblioteker.

En annen metode for å overvåke prosesser er ved å overvåke selve flyten i en prosess, for eksempel ved å bruke en debugger. Her kan man spore filendringer, I/O aktivitet og bibliotekfiler (DLL). For å benytte seg av debugger metoden trenger man kildekoden til exploiten, men hvis ikke denne er tilgjengelig kan man bruke et "reverse engineere" verktøy. IDA PRO[41] er et verktøy som gjør dette og støtter både Windows og Linux.

Det vanligste handlingsforløpet når en exploit har utnyttet en sårbarhet i en prosess, er at den injiserer en payload i prosessen. En slik payload kan utføre en handling eller returnere et shell tilbake til angriperen, for eksempel cmd.exe eller bash. Disse shellene vil vises som nye prosesser og er lette å oppdage. Det som er vanskeligere å oppdage er angrep som forblir i prosessen og forsøker å gjøre ferdig angrepet uten å skape nye prosesser eller tråder. Slike angrep kan blant annet gjøres med Metasploit[7] og såkalte "Meterpreter"- eller "The Meta-Interpreter"-scripting[42]. Med meterpreter scripting kan man scripte hva payloaden skal gjøre og det hele forsøkes kjørt i den injiserte prosessen.

For å overvåke prosesser i Windows finnes blant annet verktøyet "Process Monitor (Procmon)" fra Microsoft (Sysinternals). Procmon fungerer på den måten at den gjør et API-hook/kernel callouts mot Windows og logger aktivitet for blant annet prosesser. Procmon logger når en prosess startes, stoppes, en ny tråd blir laget og hvem som utfører dette med hvilke kommandoer. Den overvåker også stacken og flere detaljer. Procmon støtter bruk av kommandolinje og er derfor godt egnet til bruk i et automatisert system.

Som et eksempel kan Procmon startes med følgende script for å utføre en test:

Listing 5: Procmon startes for å overvåke test

```
start procmon.exe /quiet /minimized /backingfile log.pml
pause
procmon.exe /terminate
procmon.exe /OpenLog log.pml /SaveAs log.csv
```

Her startes Procmon og monitorering. Procmon logger til filen *log.pml*, som til slutt konverteres til en CSV fil. Videre stopper scriptet ved pause slik at testcaset kan bli utført før programmet avsluttes. Procmon kan også settes opp til å vente på at et program avsluttes, for så og selv avslutte loggingen.

Ved å kjøre scriptet ovenfor, samtidig som notepad.exe startes og stoppes, vil monitoreringen gi oss dette resultatet:

[11] Process ID, unikt nummer for å identifisere en prosess

Listing 6: Resultat fra start og stopp av notepad.exe som testcase i scriptet ovenfor

```
...
Explorer.EXE 1588   Process Create  C:\WINDOWS\system32\notepad.exe SUCCESS PID: 1196, Command line: "C:\
     WINDOWS\system32\notepad.exe"  TESTPC\Bruker
notepad.exe 1196    Process Start       SUCCESS Parent PID: 1588    TESTPC\Bruker
notepad.exe 1196    Thread Create       SUCCESS Thread ID: 316  TESTPC\Bruker
...
notepad.exe 1196    Thread Exit         SUCCESS Thread ID: 316, User Time: 0.0000000, Kernel Time: 0.2031250
         TESTPC\Bruker
notepad.exe 1196    Process Exit        SUCCESS Exit Status: 0, User Time: 0.0156250, Kernel Time: 0.1718750,
      Private Bytes: 860,160, Peak Private Bytes: 897,024, Working Set: 2,441,216, Peak Working Set:
     2,654,208  TESTPC\Bruker
...
```

Monitoreringen viser at Explorer.exe startet notepad.exe og at det er brukeren "Bruker" på maskinen "TESTPC" som utførte handlingen.

I Linux har man også mulighet til å se prosessaktiviteten. Blant annet finnes "Process Monitoring Linux Kernel Module"[43]. Dette er en modul for Linux-kernelen som oppretter en device under /dev/procmon som viser informasjon hver gang et program startes.

4.2.5 Datainnsamling - Nettverk

Nettverk er en ressurs som ofte kan settes i sammenheng med kvalitetssikring av exploits. Hvis målmaskinen blir infisert med ondsinnet kode er det en potensiell fare for at det vil bli forsøkt opprettet kommunikasjon med en tredjepart eller miljøet rundt. Dette kan blant annet dreie seg om opp- eller nedlasting, der hensikten er å laste ned mer kode til maskinen, eller og sende ut sensitiv informasjon som passord/brukernavn. Der hvor ormer er med i bildet, kan maskiner i samme nettverk også bli forsøkt utnyttet i et forsøk på å spre seg videre. I testmiljøet er det på ingen måte ønsket å forhindre denne type aktivitet, men heller og ta vare på det for senere analyse.

Typisk trafikk man vil overvåke med hensyn til ondsinnet kode er DNS-oppslag, http-forespørsler, og innkommende trafikk til en eventuell nyopprettet port (backdoors etc.). Overvåkning av nettverksressursen må foregå i sanntid. I likhet med prosesser er det heller ikke her hensiktsmessig å ta i bruk overflateanalyse, i og med at systemet brukes i en fryst tilstand.

Monitorering av denne typen kan blant annet la seg løse ved å benytte en honeypot[12] som for eksempel honeyd[44], eller ved hjelp av en aktiv sniffer kjørende på målmaskinen. En honeypot gir deg muligheten til å simulere både sårbare maskiner, og maskiner som tilsynelatende er "viktige" og holder sensitiv informasjon. Alt etter hva slags trafikk/-mønster man ønsker å fange opp.

En annen metode som kan monitorere trafikken, er å sette opp en maskin som logger all utgående trafikk fra testmaskinen. Dette kan for eksempel gjennomføres ved å opptre som en gateway for den aktuelle testmaskinen. Ved en slik overvåkning får man registrert all kommunikasjon, og ikke bare "angrep" som ved bruk av honeypot.

[12] Felle for å lokke til seg angripere. Kan for eksempel simulere et sårbart system for å vekke interesse hos en angriper.

Omfanget av innsamlingen kan enkelt reguleres i overvåkningen. Hvis det for eksempel ikke kan godtas at en exploit går ut på internett, trenger kun DNS-oppslag og http-forespørsler overvåkes. Ønsker man å gå dypere i selve innsamlingen kan det for eksempel settes sammen en netflow[13] som samler inn all kommunikasjon som foregår mellom to IP-adresser. Denne metoden sparer ressurser og man kan filtrere ut mye unødvendig trafikk. Hvis det er ønske om at *all* kommunikasjon blir overvåket, må selvsagt all trafikk fra testmaskinen logges. Ulempen dette medbringer er den store mengden urelevante data man sitter igjen med, og eventuell kommunikasjon fra exploiten må filtreres ut.

Det finnes i dag allerede mange gode verktøy for å overvåke nettverkstrafikk. Et kraftig verktøy som fungerer like godt i Windows som i UNIX er *TCPdump* [14]. Dette verktøyet kjører i realtime på målmaskinen, og sniffer alt man måtte ønske av nettverkstrafikk. I Unix baserer dette verktøyet seg kun på "libpcap"[45], mens i Windows benyttes "WinPcap" (omskrevet versjon av libpcap). Disse verktøyene er igjen kommandolinje-basert som gjør automatisering og scripting enklere. Man kan for eksempel dumpe DNS-oppslag fra og til host ved å bruke: "tcpdump -tttt "udp port 53". Resultatet fra denne kommandoen vises i Figur 4.

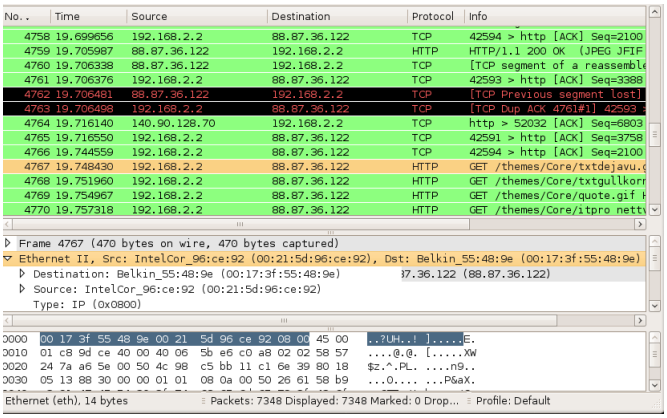
```
$ sudo tcpdump -tttt udp port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
2009-04-21 13:34:49.456485 IP p02535-k105.hig.no.45516 > inu1.inu.no.domain: 32809+ A? www.ubuntu.org. (32)
2009-04-21 13:34:49.456607 IP inu1.inu.no.domain > p02535-k105.hig.no.45516: 32809 2/0/0 CNAME[|domain]
2009-04-21 13:34:49.457613 IP p02535-k105.hig.no.36334 > inu1.inu.no.domain: 32661+ PTR? 2.32.39.128.in-addr.arpa. (42)
2009-04-21 13:34:49.457864 IP inu1.inu.no.domain > p02535-k105.hig.no.36334: 32661 2/0/0 CNAME[|domain]
2009-04-21 13:34:49.458117 IP p02535-k105.hig.no.55979 > inu1.inu.no.domain: 20805+ PTR? 38.80.39.128.in-addr.arpa. (43)
2009-04-21 13:34:49.458255 IP inu1.inu.no.domain > p02535-k105.hig.no.55979: 20805* 1/0/0 (75)
2009-04-21 13:34:49.638545 IP p02535-k105.hig.no.51653 > inu1.inu.no.domain: 15337+ A? www.ubuntu.upc.edu. (36)
```

Figur 4: Bruk av tcpdump

Wireshark[46] er en videreutvikling av Ethereal[47] og er et nettverksanalyse verktøy. Det opptrer som en pakke-sniffer på lik linje med tcpdump, men tilbyr i tillegg et intuitivt grafisk brukergrensesnitt. Wireshark er et kraftfullt verktøy som brukes av alt fra nettverksadministratorer til vanlige brukere som ønsker større innsikt i pakkestyrte nettverk.

I likhet med tcpdump og flere analyseverktøy lagrer Wireshark innsamlingen i libpcap format. I forhold til andre verktøy har Wireshark noen unike funksjoner som blant annet å sette sammen alle TCP pakkene i utvekslingen, og vise disse i ASCII, EBCDIC eller HEX-format til brukeren. I tillegg har Wireshark enorm funksjonalitet innen filtrering, noe som gjør verktøyet svært konkurransedyktig.

[13] Lytte på nettverket og samle trafikkflyt (adresser, porter/protokoller og tidspunkt)
[14] Windows utgaven heter WinDUMP



Figur 5: Innsamling ved hjelp av Wireshark

Både TCPdump og Wireshark har muligheter for å sniffe trafikk fra flere grensesnitt, noe som ytterligere forenkler prosessen dersom man bruker virtualisering (eksempelvis VMware). Hvis man ikke har mulighet for å sette opp en egen gateway, eller ikke vil sniffe trafikk direkte på målmaskinen kan man da velge å sniffe trafikk fra VMware sine egne nettverksgrensnitt (vmnet1 til vmnet8), alt ettersom hvilket nett de virtuelle maskinene er satt opp til å bruke.

Netstat er et kommandolinjeverktøy som følger med både Windows og Linux plattformen. Det er et verktøy som ofte blir brukt for å spore problemer i nettverkstrafikken. Bruksområdet i kvalitetssikringen er å spore endringer som exploiten påfører nettverkstrafikken. Ved hjelp av netstat kommandoen kan man få oversikt over antall pakker som er sendt, mottatt og kastet i systemet. En annen funksjon er at man kan få ut informasjon om de forskjellige tilkoblingene systemet har med omverdenen. En ulempe med dette verktøyet er at det må kjøres i sanntid på systemet som skal overvåkes.

```
$ netstat -ta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Addr
tcp        0      0 localhost:ipp           *:*
tcp        0      0 SL300.local:50246       by2msg301080
tcp        0      0 SL300.local:45999       mg-in-f125.:
tcp        0      0 SL300.local:36167       irc.homelien
tcp        0      0 SL300.local:35567       fx-in-f83.go
tcp        0      0 SL300.local:53184       pop.online.n
```

Figur 6: Bruk av netstat -ta kommandoen

4.2.6 Datainnsamling - Register

Datainnsamling fra registeret er en ressurs som kun gjelder for Windows-plattformen, men er allikevel inkludert med tanke på dens brukermasse og registeret sin svært sentrale del i operativsystemet. Registeret er også et kjent sted å gjemme data eller legge inn

ondsinnet kode[48]. Fra ståstedet til en dataetterforsker er det derfor nødvendig å forstå kompleksiteten og viktigheten til registeret, for og få en så god oversikt som mulig over hva som har skjedd.

Registeret inneholder informasjon som Windows kontinuerlig refererer til under normale operativsystemoperasjoner. Dette kan blant annet være profilhåndtering, hvilken type hardware som er i maskinen, hvilke applikasjoner som er installert, samt innstillinger og informasjon[49] rundt dette. I tidligere versjoner av Windows (3.11 og frem til 95) var registeret basert på tekstbaserte .ini filer for å holde denne type informasjon, men har i senere tid gått vekk fra dette, grunnet kompleksitet og optimaliseringsvanskeligheter. Som nevnt innledningsvis i rapporten har vi valt å avgrense prosjektet til den mest brukte versjonen av Windows, nemlig Windows XP[50]. I Windows XP kan registeret lokaliseres i mappene "Windows\system32\config" og C:\Documents and Settings\%USERNAME%

I begge disse lokasjonene finner man filer av type .log, .sav, .dat og noen uten benevnelse. Filene av type .sav er automatiske sikkerhetskopier av ulike deler av registeret og .log er filer med informasjon om endringer av nøkler. Filene som ikke har noe benevnelse er større deler av registeret (HKEY_LOCAL_MACHINE og HKEY_USER) som er lagret til disk. Filene kan modifiseres manuelt, men bruk av dedikerte register-verktøy er som regel mest oversiktlig. Som man kan se ved å åpne registeret med et grafisk verktøy (eksempelvis Regedit eller Regscanner[51]), så er registeret delt opp i såkalte "Hives"[52], eller *grener* på norsk. Hver gren inneholder nøkler, subnøkler og deres respektive verdier. Nedenfor vises de fem grenene til et register i Windows XP.

- **HKEY_CLASSES_ROOT (HKCR)**
  Inneholder informasjon om ulike filtype-assosiasjoner og registrering av COM-klasser som f.eks ProgID, CLSID og IID. Formålet med denne grenen er sørge for at riktig program åpnes når brukeren åpner en fil. Fra og med Windows 2000 er denne informasjonen lagret i HKLM\Software\Classes og HKCU\Software\Classes. Førstnevnte inneholder standardinnstillingene som gjelder for alle brukere på PC-en, mens sistnevnte overstyrer disse og gjelder kun for den aktuelle brukeren. Man kan si at KHCR tilbyr en visning av registeret som sammenstiller informasjonen fra disse to nøklene

- **HKEY_USERS (HKU)**
  Inneholder individuelle preferanser for hver bruker. Typisk er dette brukerprofilen som blir lastet ved innlogging. Eksempelvis brukerinnstillinger som skrivebordsoppsett, nettverksinnstillinger og applikasjonsinnstillinger.

  Tilhørende filer:
  HKU\Default: *Default, Default.log, Default.sav*

- **HKEY_CURRENT_USER (HKCU)**
  Informasjon i denne nøkkelen er bygd opp av HKEY_USERS under innloggingsprosessen. Innholdet av denne nøkkelen er derfor en kopi av undernøkkelen HKU*brukernavn*.

- **HKEY_LOCAL_MACHINE (HKLM)**
  Inneholder dataspesifikk informasjon om maskinen. Eksempelvis informasjon om maskinvare- og operativsystemdata som minne, enhetsdrivere og kontrollparametere ved oppstart.

  Tilhørende filer:
  HKLM\SAM: *Sam, Sam.log, Sam.sav*
  HKLM\Security: *Security, Security.log, Security.sav*
  HKLM\Software: *Software, Software.log, Software.sav*
  HKLM\System: *System, System.alt, System.log, System.sav*

- **HKEY_CURRENT_CONFIG (HKCC)**
  Inneholder ingen egne data, men lagrer en peker til innholdet av HKLM\SYSTEM\CurrentControlSet \Hardware Profiles\Current. Her ligger det konfigurasjonsdata om maskinvareprofilen som blir brukt av maskinen ved oppstart. Informasjonen kan enten endres her eller i HKEY_CURRENT_CONFIG. Dette for å tilby en lettere måte og aksessere dataen.

  Tilhørende filer:
  System, System.alt, System.log, System.sav og NTuser.dat (C:\Documents and Settings\username\NTuser.dat

Det finnes utallige steder i registeret hvor det kan legges inn data for å kompromittere systemet. Et eksempel på dette er vha. nøkkelen HKCR\exefile\shell\open\command\. Grenen HKCR inneholder som nevnt nøkler som blant annet har informasjon om de ulike filtypene i Windows. Denne nøkkelen bestemmer hva som skal skje i det en .exe fil kjøres. Standardverdien er *"%1" %**, men hvis denne for eksempel endres til *ondsinnetkode.exe "%1" %** så vil det resultere i at ondsinnetkode.exe kjøres hver gang en .exe fil blir åpnet. Det samme gjelder også for nøklene:
HKCR\batfile\shell\open\command
HKCR\comfile\shell\open\command

Registeret er også en utmerket kilde og benytte for å trekke ut detaljert informasjon om stort sett hva man måtte ønske. Eksempler kan være HKLM\System\CurrentControlSet \Services, som viser informasjon om Windows sine registrerte tjenester, og under hvilket navn de blir vist til brukeren. Nøkkelen HKLM\Software\Microsoft\Windows\CurrentVersion\Run er blant annet én av nøklene som viser hvilke programmer som er satt til å starte i det man logger seg på systemet. Nedenfor vises en mer utfyllende oversiktliste over de nøklene som inneholder oppstartsrutiner for både programmer og prosesser.

RunServicesOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

RunServices
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices

RunOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce

Run
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\Run

Userinit
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit

ShellServiceObjectDelayLoad
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad

For en mer detaljert liste over ulike relevante nøkler og beskrivelse, vises det til Access-Data sin oversiktsliste[53].

Alternativt kan man benytte tredjepartsverktøy som f.eks Autorunsc[54], som gir deg enda flere bruksmuligheter som eksempelvis å gi resultater i blant annet .XML og .CSV format. Dette kan være svært hensiktsmessig med tanke på analysering, sammenstilling og presentering av data i senere tid.

Selv om det her er henvist til noen systemkritiske plasseringer i registeret som er nødvendig å overvåke, så holder det ikke og bare se på disse. Likevel kan det å monitorere *hele* registeret i sanntid (dynamisk analyse) være svært krevende da det genereres flere hundre spørringer i sekundet, kun når operativsystemet går på "tomgang". For å minske mengden data, men samtidig fange opp den mest suspekte aktiviteten, kan det derfor være hensiktsmessig å fokusere på register-aktivitet som omhandler oppretting, sletting og modifisering, og heller utelukke operasjoner som spørringer, åpning, lukking etc. Et verktøy som kan gjøre nettopp dette er Regmon[55], men for nyere versjoner av Windows (XP, Vista og Windows 7) er verktøyet i stedet blitt integrert i det tidligere nevnte verktøyet, Process Monitor.

Dynamisk analyse av registeret
Et eksempel på hvordan man kan detektere skriving, endring og sletting er vist nedenfor. I eksempelet ble det laget et scenario hvor det ble opprettet en nøkkel "HKLM\Software \TestKey" med en tilhørende verdi "TestValue". Innholdet i denne verdien ble så satt til

strengen ”TestData”. Til slutt omdøpes ”TestValue” til ”TestValueRenamed”, og deretter slettet sammen med foreldre-nøkkelen ”TestKey”

Listing 7: Resultat fra Procmon som logger registerendringer i sanntid

```
...
"08:44:22,7987210","regedit.exe","1844","RegCreateKey","HKLM\SOFTWARE\TestKey","SUCCESS","Desired Access:
     Maximum Allowed"
"08:44:31,7655550","regedit.exe","1844","RegSetValue","HKLM\SOFTWARE\TestKey\New Value #1","SUCCESS","Type:
     REG_SZ, Length: 2, Data: "
"08:44:39,3659699","regedit.exe","1844","RegSetValue","HKLM\SOFTWARE\TestKey\TestValue","SUCCESS","Type:
     REG_SZ, Length: 2, Data: "
"08:44:49,7877123","regedit.exe","1844","RegDeleteValue","HKLM\SOFTWARE\TestKey\New Value #1","SUCCESS",""
...
"08:45:05,3761243","regedit.exe","1844","RegSetValue","HKLM\SOFTWARE\TestKey\TestValue","SUCCESS","Type:
     REG_SZ, Length: 14, Data: TestData"
...
"08:45:28,4978554","regedit.exe","1844","RegSetValue","HKLM\SOFTWARE\TestKey\TestValueRenamed","SUCCESS","
     Type: REG_SZ, Length: 14, Data: TestData"
"08:45:33,5897414","regedit.exe","1844","RegDeleteValue","HKLM\SOFTWARE\Truls\TestValue","SUCCESS",""
...
"08:46:40,4267419","regedit.exe","1844","RegDeleteValue","HKLM\SOFTWARE\Truls\TestValueRenamed","SUCCESS",""
"08:46:48,1501531","regedit.exe","1844","RegDeleteKey","HKLM\SOFTWARE\TestKey","SUCCESS",""
```

Som man kan se ut i fra eksempelet ovenfor så er det operasjonene RegCreateKey, RegSetValue og RegDeleteValue som er gjengangere, og det er nettopp disse man bør se etter ved senere bruk. Det at man ser fullt klokkeslett, navn på prosess som utfører endringen, og om utfallet er suksessfullt eller ikke, er mer en nok informasjon for å kunne analysere resultatet på en troverdig måte.

Det er også mulig å følge *én* bestemt prosess, for å se hva slags aktivitet akkurat denne har mot registeret. Et verktøy som kan muliggjøre dette er for eksempel ”RegFromApp”[56]. Nedenfor er det vist et lite eksempel på hvordan man kan starte monitorering av programmet Firefox.

```
RegFromApp.exe /RunProcess ''C:\Program Files\Mozilla\Firefox.exe'' /StartImediately 1 /AutoSave
     registerlogg_firefox.log
```

Ved å hente ut LastWriteTime verdien[57] til en bestemt nøkkel, kan man se tiden når siste endring ble utført. Dette er en verdi som alle registernøkler har, og kan sammenlignes med MAC-tidsstempler på filer. Oppdateringen av denne verdien skjer automatisk og fortløpende i det Windows detekterer en endring.

Overflateanalyse av registeret

For å først se på en tilnærmet statisk metode for å detektere endringer i registeret kan man dumpe hele registeret til tekstfiler ved hjelp av programmet reg.exe som er standard i Windows. Man kan også gjøre dette ved bruk av mer dedikerte verktøy som HiJackThis[58], som kan startes før og etter kjøring av exploit for å avdekke eventuelle differanser. Da dette programmet er utviklet med fokus på brukervennlighet er det blitt laget et eksempel på hvordan man kan gjøre denne prosessen så lite interaktiv som mulig:

Listing 8: Bat-script som ser på endringer før og etter eksekvering av exploit

```
@echo off

c:\tmp\HiJackThis.exe /silentautolog /ihatewhitelists && ping 127.0.0.1 -n 5 > nul
rename c:\tmp\hijackthis.log before_exploit.txt

:: < kalling av andre programmer >
:: < Starte exploit >

c:\tmp\HiJackThis.exe /silentautolog /ihatewhitelists && ping 127.0.0.1 -n 5 > nul
rename c:\tmp\hijackthis.log after_exploit.log

:: < diff av before_exploit og after_exploit.log >
```

Det som menes med en "tilnærmet statisk analyse" i første avsnitt er at man ved bruk av disse metodene ikke jobber med et dødt filsystem, men i stedet innhenter denne informasjonen i et fullt operativt system. Innhentingen av informasjon utføres riktignok før og etter utført exploit, men troverdigheten svekkes med tanke på at eventuell ondsinnet kode kan påvirke resultatene.

For en mer fullstendig utførelse av statisk analyse kan man igjen for eksempel benytte funksjonene til VMware for å montere filsystemet, kopiere ut registerfilene (Hives) fra eksempelvis %systemroot%\System32\config og %SYSTEMROOT%\Documents and Settings\%Brukernavn\. Dissen filene er binærdata, og må derfor parses med dedikerte verktøy som for eksempel DumpHive[59] eller vha. modulen "Parse::Win32Registry"[60]. Begge disse har mulighet for å analysere "rådataen" fra de lagrede registerfilene og presentere de i leselige tekstformater, som igjen senere kan benyttes til sammenligning.

4.2.7 Datainnsamling - Minne

Ett av problemområdene som ble nevnt under innsamling av filaktivitet tok for seg hvordan en exploit sin payload kunne unngå aktivitet mot disk ved og kun operere i minnet. For å vanskeliggjøre og/eller forhindre denne type aktivitet finnes det flere metoder som f.eks. ulike typer HIPS systemer[15]. Disse verktøyene er tilgjengelige for de fleste plattformer og tilbyr blant annet systemkall-analyse, implementasjon av ikke-eksekverbare stacker og ASLR[16], som vilkårlig endrer plasseringen til heap, stack og andre nøkkeldata i prosessers adresseområde. Det å beskytte seg mot denne type angrep/aktivitet er én ting, men og spore/finne igjen relevante data i minnet blir noe annet. En av tingene som vanskeliggjør dette er blant annet at det kreves forskjellig fremgangsmåte for ulike typer operativsystem. Den mest egnede måten å analysere minnet på er ved hjelp av statisk analyse, selv om det også finnes metoder for å hente ut minnet i "sanntid".

I eldre versjoner av Windows (frem til Windows server 2003 SP1) er det to kjente metoder som ofte blir brukt for å hente ut minnet i sanntid; den ene er kopiering av minnedumpen \\.\Device\PhysicalMemory[61] ved hjelp av dd eller netcat, og den andre er fremprovosering av systemkrasj[62] ved å endre en register-nøkkel. Sistnevnte gjøres for å generere

[15] Host-based Intrusion Prevention Systems
[16] Address space layout randomization

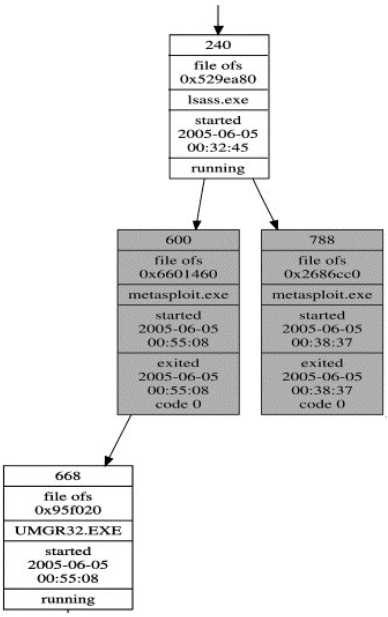
en fysisk minnedump, slik at man kan få tak i filen *Memory.dmp*. Begge metodene har sine fordeler og ulemper, men vil ikke bli gått videre inn på her.

I et virtuelt testmiljø (eksempelvis VMware) er det hensiktsmessig å benyte en statisk metode for minneanalyse med tanke på de lett tilgjengelige minnedumpene .vmem og .vmss. Dette er filer som tilgjengeliggjøres i det man midlertidig stanser gjesteoperativsystemene, og som videre kan kopieres ut og analyseres med ulike verktøy for å finne ønskede data. Disse dataene kan blant annet være:

- Kjørende prosesser
- Skjulte prosesser
- Terminerte prosesser
- Nettverksforbindelser
- Åpne nettverksporter
- Åpne filer
- Lastede moduler (eksempelvis .DLL filer)
- Filer åpnet av prosesser

Det er flere verktøy tilgjengelig på markedet som kan være behjelpelig med å finne deler av informasjonen nevnt ovenfor. Eksempler på dette er de anerkjente verktøyene PTfinder[40] og Volatility fra Volatile Systems[63]. Ingen av programmene er ment som fullverdige analyseverktøy for minnet, men gjør at man kommer godt på vei og får dekket de grunnleggende behovene. Felles for de begge er at de har åpen kildekode og er lisensiert under GNU General Public Licence.

PTfinder er skrevet i Perl og tar for seg analysering av prosesser og tråder, og er kompatibel med de fleste dump-filformater. Programmet er vanligvis å se som et tekstbasert grensesnitt, men som Andreas Schuster viser i en av sine presentasjoner[64] er det mulig å bruke PTfinder sammen med grafiske verktøy som for eksempel ZGRviewer[65] og GraphViz[66], for å se handlingsgangen til utvalgte prosesser på en mer intuitiv måte. Eksempelet i presentasjonen tar blant annet for seg "grafisk sporing" (se figur 7) av den kjente LSASS-exploiten[67], samt avdekking av skjulte rootkits. Den grafiske visningen av parent-child forholdet for prosessene lar seg gjøre ved bruk av PID- og PPID-informasjon, gitt i EPROCESS strukturen.



Figur 7: Grafisk visning av prosesser fra minnedump ved hjelp av PTfinder og ZGRviewer

Som man kan se fra figur 7, så oppretter prosessen LSASS.exe (Local Security Authority Subsystem) en annen prosess navngitt UMGR32.EXE. Siden LSASS normalt sett ikke skal starte andre prosesser, kan man allerede her begynne å mistenke unormal aktivitet.

Det andre verktøyet som er verdt å merke seg innen statisk minneanalyse er Volatility. Volatility er et rammeverk med en samling av verktøy skrevet i Python. På lik linje med PTfinder er dette også fri programvare, og har muligheter for å analysere en rekke dump-filer, inklusive krasj-dumper som nevnt innledningsvis. Nedenfor er det listet opp de mest kjente og brukte modulene i programmet.

connections	Print list of open connections
connscan	Scan for connection objects
datetime	Get date/time information for image
dlllist	Print list of loaded dlls for each process
files	Print list of open files for each process
modules	Print list of loaded modules
pslist	Print list of running processes
psscan	Scan for EPROCESS objects
sockets	Print list of open sockets
thrdscan	Scan for ETHREAD objects
vadinfo	Dump the VAD info
vadwalk	Walk the vad tree

For å gi et inntrykk av hvordan verktøyet fungerer er det nedenfor vist et eksempel på bruk og resultat fra modulen psscan, med en Windows XP minnedump:

python volatility.py psscan -f ../Victim_XP-Snapshot2.vmem

Listing 9: Eksempel på bruk av modulen *psscan*

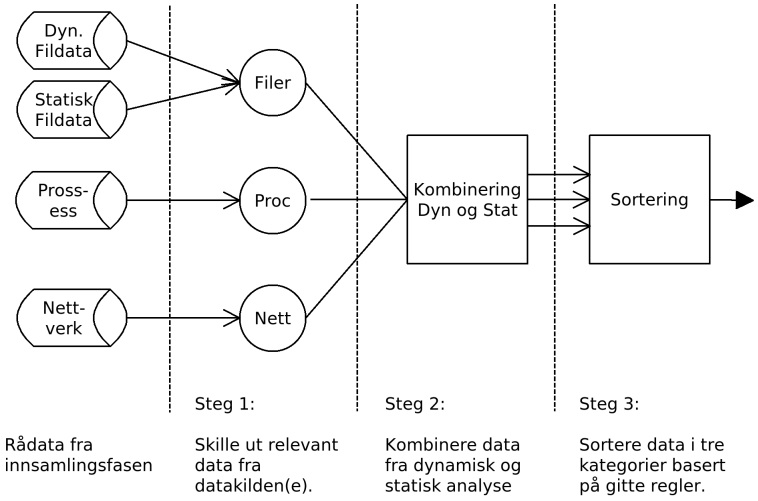
```
No.  PID    Time created             Time exited              Offset     PDB        Remarks
---- ------ ------------------------ ------------------------ ---------- ---------- ----------------

   1      0                                                   0x00544640 0x00039000 Idle
   2    816 Mon Apr 20 09:42:31 2009 Mon Apr 20 09:44:14 2009 0x018c78f0 0x043e5000 VMwareUser.exe
   3    184 Mon Apr 20 09:42:29 2009 Mon Apr 20 09:44:16 2009 0x018d9618 0x03a0d000 VMwareService.e
   4   1132 Mon Apr 20 09:47:11 2009 Mon Apr 20 09:54:20 2009 0x018fe020 0x0c3c4000 msoobe.exe
   5   1916 Mon Apr 20 09:45:01 2009                         0x019118b8 0x0a2f3000 VMwareService.e
   6   1732 Mon Apr 20 09:44:55 2009                         0x01939020 0x09749000 msmsgs.exe
   7   1724 Mon Apr 20 09:44:55 2009                         0x0193c640 0x09712000 VMwareUser.exe
   8   1556 Mon Apr 20 09:44:54 2009                         0x01967020 0x0870f000 explorer.exe
  ......
  19   1420 Mon Apr 20 09:22:04 2009 Mon Apr 20 09:44:16 2009 0x01a0c250 0x0884c000 spoolsv.exe
  20    360 Mon Apr 20 09:44:49 2009                         0x01a2baa0 0x03170000 smss.exe
  21    648 Mon Apr 20 09:14:48 2009                         0x01a6cb68 0x0a100010 lsass.exe
  22      4                                                  0x01bce800 0x00039000 System
```

4.3 Dataanalyse/Filtrering

Ved endt datainnsamling sitter man som oftest igjen med store mengder rådata som skal filtreres og analyseres. Selv om man i innsamlingsfasen kun forsøker å fange opp så relevant data som mulig, er ikke dette alltid like lett med tanke på troverdighet og opprettholdelse av testens kvalitet. Velger man å tilspisse datainnsamlingen for mye står man i fare for å miste informasjon av relevans, mens man ved innhenting av for mye data risikerer og få for mange falske positiver. Det er derfor viktig å finne en balansegang.

I datainnsamlingen benyttes det ofte flere forskjellige verktøy, noe som gir rådata i mange formater. En utfordring i dataanalysen vil være å kunne hente inn og konvertere disse dataene til et felles format.



Figur 8: Eksempel på dataanalyse

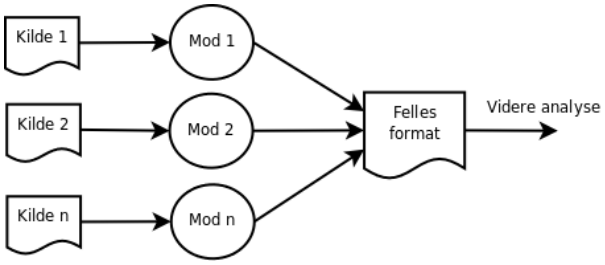
I figur 8 viser vi en mulig måte å løse disse problemstillingene. Her har vi valgt å dele opp analysen i tre moduler:

- Steg 1 - Skille ut relevante data
- Steg 2 - Kombinere overflate- og dynamisk analyse
- Steg 3 - Sortering i forskjellige kategorier

4.3.1 Steg 1 - Skille ut relevante data

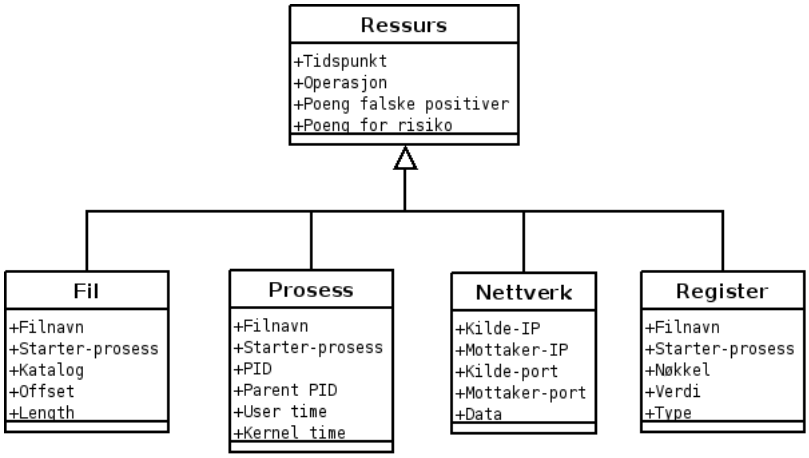
Dette steget i dataanalysen er laget for å løse utfordringen med rådata fra flere forskjellige kilder. Som tidligere nevnt hentes det inn data ved hjelp av dynamiske og statiske teknikker. Dette gjøres på forskjellige stadier i analysen, og med en rekke forskjellig programvare. Ettersom vi i størst mulig grad prøver å benytte oss av ferdig/åpen programvare, vil formatet på dataene som oftes være forskjellig fra program til program. Rådata som samles inn vil i de fleste tilfeller være lagret som ASCII i en eller flere filer. Det kan også forekomme at de er lagret som binærdata i en form for database, eller presenteres i et GUI. Det er en forutsetning at resultatet kan leses maskinelt, derfor vil man som oftes styre unna lukkede formater.

En mulig løsning for å benytte data fra flere kilder vil være og ta i bruk en modulær arkitektur. Det vil si at man skiller ut funksjonaliteten for innlesing fra de forskjellige datakildene i egne moduler. I figur 9 ser du at hver datakilde har en egen modul som står for innlesningen. Resultatet fra hver modul blir gjort om til et felles format og sendes videre for analyse.



Figur 9: Data fra flere kilder til felles format

Et av de første stegene for en slik tilnærming er å etablere et felles dataformat for hver ressurs. For filer vil dette typisk være filnavn, operasjon, "prosess som utførte operasjon" etc, mens for nettverk er ip-adresser og protokoller interessant. I figur 10 er det et UML-diagram med oversikt over ressurser og hvilke egenskaper disse skal ha. Egenskapene i "Ressurs" er felles for alle ressurser og arves derfor ned til alle undernoder.



Figur 10: Ressurser og egenskaper

I tillegg til at man grupperer modulene etter hvilke ressurser de henter data fra, må de også grupperes etter om de leser inn data fra en statisk eller dynamisk kilde. Grunnen til dette er at man ønsker et dobbel sett med data hentet inn med forskjellige teknikker, som i sin tur kan sammenlignes for å gi et mer pålitelig resultat. Dette vil bli bedre beskrevet nedenfor.

4.3.2 Steg 2 - Kombinere overflate- og dynamisk analyse

Alle data som samles er enten hentet fra en statisk kilde eller en dynamisk kilde. Med andre ord er det ene datasettet samlet inn mens operativsystemet kjører, mens det andre

datasettet er samlet når det er fryst. I utgangpunktet kan det virke unødvendig å hente inn samme data på to forskjellige måter, men sannheten er at dette kan bidra til å sikre testens integritet.

Hvis man skulle velge enten dynamisk eller statisk analyse, ville sannsynligvis de fleste valgt dynamisk, ettersom denne tilbyr mest informasjon. En dynamisk analyse må som sagt kjøres på et aktivt system for å kunne samle inn data. Dette er grunnen til at teknikken kan fange inn så mange detaljer, men det er også den største ulempen, sett fra et sikkerhetsperspektiv. Hva er det som stopper ondsinnet programvare fra å oppdage og forstyrre en analyse på et kjørende system? Dessverre veldig lite. Men det er her den statiske analysen kommer inn i bildet. Ved å kjøre samme analysen på et "dødt" system har ikke den ondsinnede programvaren mulighet til å forstyrre testen. Hvis vi i tillegg sammenligner resultatene fra begge testene, vil det være mulig å avdekke eventuelle integritetsproblemer.

Ved sammenligning av overflate og dynamiske innsamlede data må man ta hensyn til at man ikke får like mange detaljer ved bruk av statiske som dynamiske teknikker. Dette er på grunn av, at statiske teknikker ikke får med seg mer enn den siste endringene. Det vil derfor gi best resultat hvis man går gjennom alle statiske data og sammenligner med det dynamiske datasettet.

4.3.3 Steg 3 - Sortering i forskjellige kategorier

For å kunne sortere exploits til egnede kategorier er det viktig å ha fastsatt kriteriene for kvalitetssikringen (se punkt 3.9). Kriteriene bør settes inn i en policy som er i tråd med formålet for penetrasjonstestingen. Nedenfor kommer det forslag på en mulig kategorisering ved presentasjon av testresultatene.

Det første presentasjonsnivået kan være et sammendrag som inneholder de viktigste resultatene som har kommet frem i testen. I sammendraget bør det leggest vekt på å vise resultatene ved hjelp av lettfattelige figurer, som inneholder de viktigste endringene. Filer og prosesser som har blitt opprettet/terminert, endringer i registeret og nettverkstrafikk er eksempler på slik aktivitet.

Presentasjonsnivå to er "den gylne middelvei" i kvalitetssikringen. I tillegg til sammendraget kan denne presentasjonen inneholde alle endringer som påføres målsystemet av exploiten. Denne vil være mer nøyaktig og gir en bedre oversikt over endringene, enn sammendraget. Ved denne presentasjonen fjernes de falske positivene fra rådataen og all data som ikke blir filtrert ut av analysen forespeiles brukeren.

Det tredje og siste presentasjonen bør inneholde all rådata som har blitt samlet inn ved overvåkningen. Rådataen kan kunne presenteres både kronologisk og kategorisert på hvilken ressurs endringen tilhører. Denne presentasjonen er for spesielt interesserte, som gjerne vil undersøke endringene grundigere. Størrelsen på loggen kan variere, men ofte vil dette bli en stor og uoversiktelig logg.

4.3.4 Falske positiver

Når de første datamaskinene kom, hadde man begrenset regnekraft og det ble kun kjørt såkalte batchjobber. En batchjobb var en større jobb som kjørte alene på datamaskinen, og hadde alle ressursene tilgjengelig. Hadde man flere jobber, måtte man gjøre seg ferdig med en jobb før man startet den neste. Slik er det ikke lenger i dag. Man har fått datamaskiner med stor regnekraft, og så godt som alle moderne operativsystemer benytter seg av multitasking, det vil si å kunne kjøre mer enn én prosess samtidig. Det er i det hele tatt mye som skjer under panseret på et operativsystem - selv når brukeren ikke gjør noe. For å ta et eksempel har Windows XP en prosess som holder oversikt over hvilke ikoner som er minst brukt på skrivebordet, mens mange utgaver av Linux daglig oppdaterer søkeindeksen for filer.

Denne ekstra aktiviteten er en utfordring når man skal overvåke en exploit. Hva er reell aktivitet og hva er bakgrunnsstøy? Ved å monitorere aktiviteten i operativsystemet når det "går på tomgang", er det mulig å se noe av det som skjer. Nedenfor vises noen eksempler på aktivitet i Windows XP ved noen minutter med "tomgangskjøring" og vanlig navigering:

1. **WINDOWS\system32\wbem\Logs\WinMgmt.log is added**
   Loggfil i WBEM (Web-Based Enterprise Management). Er en kjernekomponent i Windows XP - Vedlikeholder seg selv

2. **WINDOWS\Prefetch\MSTINIT.EXE-39813B24.pf is deleted**
   Prefetcher er en Windows-komponent som hjelper til med å starte programmer raskere

3. **System Volume Information\restore (7D7B84AB-B36C-4000-B178-E456BA37498F) /RP1/A0001017.ini is added**
   Mappe som inneholder diverse systeminformasjon (System Restore points, Distributed Link Tracking Service databases, Content Indexing Service, Information used by the Volume Snapshot Service)

4. **RECYCLER\S-1-5-21-299502267-287218729-839522115-1003\Dc1.txt is added**
   Søppelkurven i Windows

5. **Documents and Settings\victim\Local Settings\History\History.IE5 \MSHist 012009012820090129\index.dat is added**
   Endret brukerinnstillinger på den innloggende brukeren

6. **Documents and Settings\victim\Recent\MSN.lnk is added**
   Automatisk opprettet link til sist brukt applikasjon

Selv om mange av eksemplene ovenfor helt klart er aktivitet skapt av operativsystemet, er det også en viss risiko ved å forkaste dem som falske positiver. I noen tilfeller kan "strøfiler" som dette være viktige spor etter ondsinnet kode.

En metode for å håndtere falske positiver er å lage et flitreringssystem og et poengsystem. Filtreringssystemet fjerner kjente falske positiver fra resultatet. Poengsystemet gir ulik poengsum til endringene som ligger i loggen. Dette er samme prinsippet som innen intrusion detection-systemer. Hvis sannsynligheten for at en endring er støy fra operativsystemet gis det få poeng. Hvis det derimot er stor sannsynlighet for at endringen kan være utført av exploiten, gis det flere poeng. Endringene kan dermed presenteres etter antall poeng slik at brukeren kan gå inn å undersøke de falske positivene. I og med at exploits kan være skrevet for å eksekveres i prosesser som kjører til vanlig, er det viktig med en grundig undersøkelse for å være sikker på exploiten.

Hvis man ukritisk fjerner alle antatte falske positiver, kan man fort ende opp med å skape falske negativer. Utfordringen da er å bevare alle spor etter den ondsinnede koden, uten og "drukne" brukeren i data. En mulig løsning på dette er å kategorisere dataene etter hvor relevante de er, istedenfor å fjerne data man antar er urelevante.

4.3.5 Håndtering av loggfiler

Å kvalitetssikre exploits basert på analysering av loggfiler stiller store krav til integritet. Blir loggfilene modifisert (eksempelvis av forsvarsmekanismer til ondsinnet kode) kan dette i verste fall resultere i feilaktig godkjennelse, og hele poenget med kvalitetssikring blir borte. To metoder vi ser på som delvis usikre på dette området er 1) logganalyse direkte på målmaskin, og 2) kopiering av loggfiler til en eventuelt analysemaskin *etter* utført test. Ved bruk av disse alternativene kan man ikke være sikker på om loggfilene er blitt endret underveis. Nedenfor er det vist noen metoder for å løse utfordringene med tanke på integritet og konfidensialitet.

Overføring av logg i sanntid
Denne metoden sørger for at data ikke blir liggende på målmaskinen. I det aktivitet blir registrert på målsystemet, sendes dette fortløpende til en annen maskin (eksempelvis analysemaskin eller loggserver) for videre prosessering. For å gjennomføre dette kan man ta i bruk "netcat"[17] [18], et lite verktøy som kan lese og skrive data over nettverksforbindelser. Verktøyet bruker TCP/IP-protokollen, og som vist i listingen under ser man at verktøyet er lett å ta i bruk, samtidig som det er svært kraftig.

[17]Netcat for Linux: http://netcat.sourceforge.net/
[18]Netcat for Windows: http://joncraton.org/files/nc111nt.zip

Listing 10: Oppstart av netcat på analyse-maskin

```
nc -l -p 3333 >> loggfil.log
```

Netcat (nc) startes her i en lyttende tilstand (-l), og blir satt til å taimot tilkoblinger på port 3333. I det filer/data blir sendt til denne porten blir det fortløpende lagt inn i loggfil.log.

Listing 11: Netcat - Etablering av forbindelse (UNIX)

```
tail -f datainnsamling.txt | nc logg-server-ip 3333
```

Kommandoen "tail -f" i Linux brukes blant annet for å følge endringer som blir lagt til i nye filer. Endringene blir i utgangspunktet sendt til stdout, men siden det er ønskelig å sende dataene over til analysemaskinen så sendes dette gjennom et "rør" og videre til netcat.

På loggmaskinen kan man nå fortløpende se aktiviteten som blir logget på målmaskinen. En klar ulempe med dette eksempelet er at hvem som helst kan sende data til loggserveren på lik linje som målmaskinen, og på den måten obfuskere resultatet. For å løse dette trenger man en form for autentisering og/eller krypteringsløsning. Flere verktøy er laget for dette formålet, eksempelvis Cryptcat[19](Kryptert utgave av Netcat) og SSH. Sistnevnte er mest utbredt, og har derav flest muligheter. Ved å sette opp en SSH-tunnel mellom partene har man allerede dekket begge kravene om autentisering og kryptering (konfidensialitet).

Listing 12: Overførsel av logg i sanntid vha. SSH tunnelering

```
SSH -f -N -L 1234:localhost:3333 test@analyse-maskin nc -l -p 3333 >>
    loggfil.log; tail -f datainnsamling.txt | nc localhost 1234
```

Essensen i dette eksemplet er parameteren "-L" til SSH, som etablerer en sikker tunnel gjennom "localhost:1234" til målmaskinen.

Videre vil bruk av PKI (Public Key Infrastructure), eller andre løsninger med sertifikater opp imot SSH tjeneren gjør at man slipper jobben med å manuelt skrive inn passord. Dette er noe som ytterligere forenkler prosessen og samtidig øker sikkerheten. Som siste ledd i verifisering av loggfilene vil det også være hensiktsmessig å avdekke eventuelle feil i overføringen ved å differensiere loggfilen på målmaskinen mot loggfilen på analysemaskinen.

4.4 Presentasjon

I rammeverket skal resultatene fra datainnsamlingen kunne presenteres på en oversiktelig og lettlest måte. Presentasjonen skal kunne brukes til både å gi en kort oversikt over de

[19]Cryptcat for Linux og Windows: http://sourceforge.net/projects/cryptcat/

viktigste funnene, i tillegg til en mer detaljert logg. I og med at rapporten skal kunne brukes av både sikkerhetseksperter og ledere, må rapportene derfor tilpasses forskjellige brukergrupper.

For å presentere rapporten kan det brukes forskjellige teknikker og formater. Resultatet kan blant annet gis til brukeren ved en internettside, der rapporten forekommer i for eksempel HTML/PHP format, eller det kan være at rapporten blir vist direkte som en tekstfil. Sistnevnte er lite gunstig i og med at strukturen fort kan bli uoversiktelig, og grensesnittet til brukeren blir heller ikke egnet for figurer eller diagrammer. En annen metode er ved bruk av XML. Dette er et merkespråk som brukes til å transportere og lagre data. I utgangspunktet er ikke XML så mye bedre en vanlige tekstfiler, men ved bruk av blant annet stylesheets, så kan XML fort bli lettlest og oversiktelig.

En mulig løsning for å presentere rapporten er å kategorisere dataene etter detaljnivå. Her har vi vist en måte på hvordan man sorterer dataene i tre ulike kategorier:

1. **Sammendrag**
   Første kategori er et sammendrag av resultatene hvor man for eksempen ser hvor mange filer som er opprettet eller slettet, hvor mange prosesser som er startet eller avsluttet etc. Dette sammendraget kan være en fin indikator for å se om det har vært mye eller lite aktivitet.

2. **Teknisk nivå**
   Her vises alle data, utenom de som antas å være falske positiver. Dette kan være "Tomgangstrafikk" fra operativsystemet som for eksempel antivirus, caching etc. Hvilke data som skal fjernes, bestemmes ut i fra gitte regler. Disse reglene bør også være mulig å endre for brukeren siden man etter tid får erfaring med hvilke data som er relevante.

3. **Rådata**
   Den siste kategorien viser alle data som er samlet inn. På denne måten vil brukeren har mulighet til å grave seg ned i alle detaljer, hvis dette er ønskelig for en dypere analyse.

Man kan selvfølgelig velge å operere med så mange kategorier man ønsker, men som sagt er det viktig å finne balansegangen mellom å gi brukeren for mye eller for lite informasjon.

En webside har gode tilpasningsmuligheter, og er en god løsning for eksempelvis å kunne presentere resultatene dynamisk. Nedenfor vil det fremstilles forskjellige metoder og teknikker som kan brukes, etterfulgt av teknikkene vi har valgt å bruke i eksperimentet.

I og med at det finnes mange muligheter ved bruk av websider, har gruppen vurdert ulike løsninger og tanker rundt presentasjonen på denne måten. En av de første figurene vi vurderte etter oppsummering av eksperimentet, var å lage en "risikograf". Dette ville være en illustrasjon som viste en oppsummering av alle endringene exploiten påførte systemet. Bakgrunnen for risikografen var å gi brukeren et bilde av oppførselen til exploiten. Ved

å gi poeng til hver endring ut ifra et regelsett som er satt for exploiten (Se analyse av data 4.3), kan dette illustrere hvor "kritiske" endringer som påføres systemet.

En annen teknikk som ofte er mye brukt i presentasjoner er tidslinjer. Dette visualiserer endringene over tid, i tillegg til at flere resursser kan legges inn i samme figur. Ved å gjøre tidslinjen dynamisk med for eksempel Java eller Adobe Flash kan tidslinjen endres etter brukerens ønske. Dette medfører at brukeren kan velge et tidsintervall som skal belyses eller følge en bestemt ressurs under hele eksekveringen. For å kunne bevege seg inn i en dypere beskrivelse av endringene, kan brukeren klikke seg inn til viktige hendelser i eksperimentet og få listet ut mer informasjon om hendelsen.

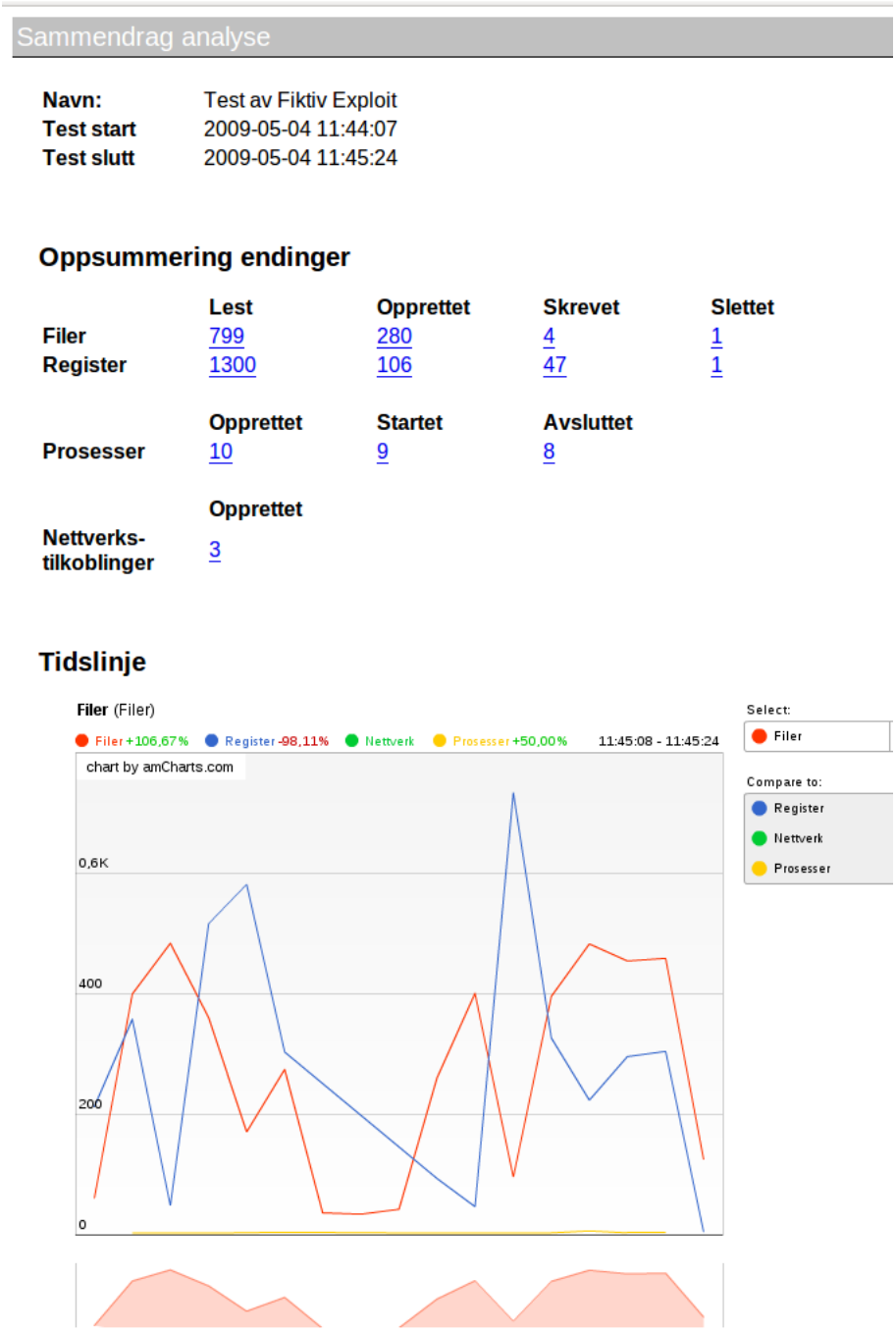
For å kategorisere endringene kan man bruke et poengsystem (kap 4.3.4). Ved å vise eller fjerne endringer på bakgrunn av antall poeng hver endring har fått tildelt, kan man få en oversikt over de mest aktuelle endringene. Å lage en topp 10 liste over endringer kan også være en god løsning for og få en oversikt over exploiten.

Det tekniske nivået og selve loggen fra eksperimentene skal også kunne presenteres for brukerne. Utfordringen med disse nivåene er at de presenterer store mengder data, så det er viktig å benytte oversiktelige metoder. Ofte benyttes det kronologisk sorterte lister som er kategorisert på forskjellige ressurser. Websider bruker ofte fanebaserte løsninger, hvor brukeren kan klikke seg nedover i materien.

Det å kunne sortere ut støy og data som ikke er relevant, er svært vesentlig for bruken av rammeverket. For at rammeverket skal benyttes, må denne funksjonaliteten være enkel og intuitiv. Ved å legge inn filtreringsfunksjoner i sammendraget har brukeren mulighet til å sortere ut informasjonen til og gjelde en valgt ressurs, eller velge resultater fra en bestemt tidsperiode. I tillegg kan det legges inn en søkefunksjon for å finne bestemte endringer. Hvis man ønsker å verifisere endringer som allerede er dokumentert, kan man for eksempel søke opp en filendring fra dokumentasjonen for og verifisere at denne har blitt registert. Dette er funksjoner som gjør at man kan filtrere ut en del falske positiver, slik at brukeren har mindre data og forholde seg til.

Det er viktig å huske på at problematikken rundt falske positiver er utfordrende og kompleks. Det å karakterisere en endring som en falsk positiv, kan resultere i tap av viktig informasjon i testingen. Ofte blir exploitene laget for å utnytte program som til vanlig brukes av operativsystemet. Ondsinnet kode kan for eksempel legges inn i oppstartsrutiner til maskinen, som medfører at koden kjøres neste gang brukeren logger på.

Vi har valgt å bruke en dynamisk webside for og vise resultatet. Dette gjør det enkelt å vise resultatene på en oversiktlig måte samtidig som man kan velge forskjellige nivåer. Bakgrunnen for valget av denne metoden er mengden av data som genereres, ønsket om oversiktlig og intuitivt brukergrensesnitt, og mulighetene for å tilpasse presentasjonen så mye man vil. I tillegg gir dette oss et stort spillerom til bruk av forskjellige figurer som i tillegg kan opptre dynamisk.



Figur 11: Presentasjonsnivå 1 - Sammendrag

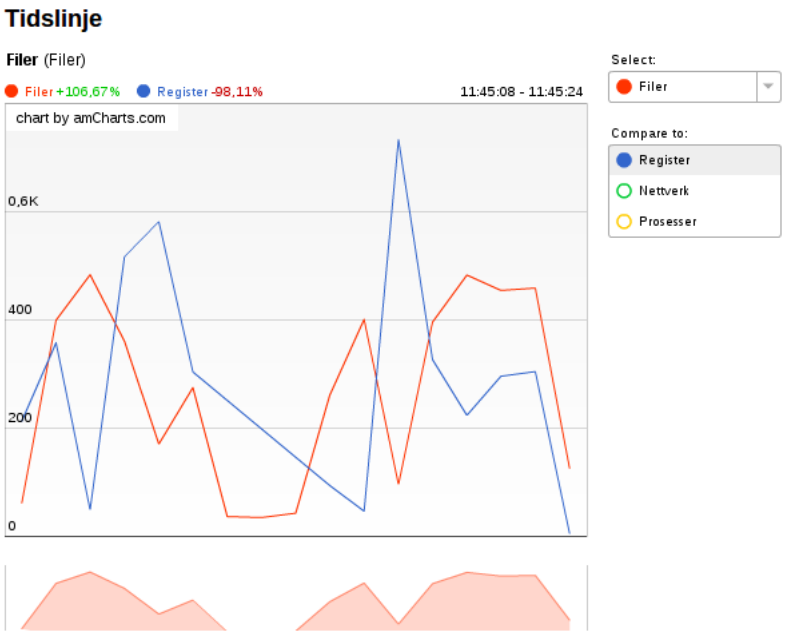
Som man kan se på figur 11, kommer det først faktaopplysninger om testeksperimentet og selve exploiten, med påfølgende oppsummeringen av endringene på de forskjellige

ressursene. For å aksessere nivå 2, med et mer teknisk nivå, kan man klikke på antallet endringer i ønsket ressurs. Figur 12 viser hvordan det dette tekniske nivået ser ut. Her vises resultatene kategorisert på tid, og hvilket program som påførte endringene.

Dato	Filnavn	Startprogram
2009-05-04 11:45:13	C:\fil2.txt	cmd.exe
2009-05-04 11:45:13	C:\fil2.txt	cmd.exe
2009-05-04 11:45:13	C:\WINDOWS\system32\fiktiv.txt	cmd.exe
2009-05-04 11:45:13	C:\WINDOWS\system32\wbem\Logs\wmiprov.log	wmiprvse.exe

Figur 12: Opplisting av endringer nivå 2, filer skrevet

Ved å ha en dynamisk tidslinje i sammendraget visualiseres endringene på de forskjellige ressursene. Som man ser av figur 13 kan man selv velge hvilke ressurser som skal vises. I dette eksempelet har vi valgt å visualisere både filer og registery. I tillegg kan man zoome inn på viktige hendelser (peeks i grafen) for å kunne liste ut endringene i valgt tidsrom. Dette medfører at man kan snevre inn eksekveringen for å luke ut falske positiver og støy fra operativsystemet.



Figur 13: Oversiktsbilde og tidslinje

En oppgave som bør vurderes og arbeides videre med er å forbedre måten man kan tilpasse forskjellige regelsett og spesialtilpasse filtrene på, slik at presentasjonteknikkene

blir bedre. Det å kunne lage flere regelsett gjør at bedrifter kan tilpasse oppsettet ut ifra hvilke exploiter man benytter seg av, og til hvilke formål de skal brukes er viktig.

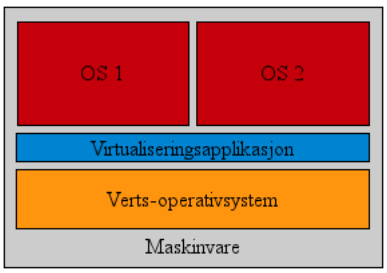
I dette kapittelet har vi valgt å fokusere mest på presentasjon av data samlet inn fra programmer som Procmon, Afick, osv. Dette er programmer som genererer data som passer bra til maskinell lesing, men som må bearbeides for at et menneske skal få nytte av det. En annen metode som er verdt å nevne er bruke programvare som lager menneskevennlig data direkte. Et eksempel på et slikt program er Snort - en IDS. Utføres det et angrep som Snort plukker opp, genereres det automatisk en beskjed til brukeren. Data som dette kan presenteres direkte til brukeren, og kan være et godt supplement til presentasjonen nevnt ovenfor.

4.5 Testmiljø

Som testmiljø har vi valgt å benytte et virtuelt miljø. Et virtuelt miljø kan raskt settes opp og være klart til testing av exploits, samtidig som det også er enkelt å tilbakestille slik at det er klart for en ny test.

4.5.1 Mulige testmiljø

Virtualisering vil si å dele maskinvare mellom flere operativsystem der operativsystemene kan kjøres uavhengig av hverandre. Det finnes flere typer virtualisering, blant annet plattformvirtualisering, applikasjonsvirtualisering og desktopvirtualisering. Siden vi skal monitorere et helt operativsystem vil det være mest nyttig å bruke plattformvirtualisering og fullvirtualisering (se figuren 14). Fullvirtualisering vil si at vi har et vertsoperativsystem som kjører normalt på maskinen. Her installerer vi virtualiseringsapplikasjonen hvor det igjen kan installeres flere gjesteoperativsystemer. Et gjesteoperativsystem kjører direkte i det virtuelle miljøet, og det virtuelle miljøet emulerer hardware for operativsystemet. På denne måten kan vi isolere gjesteoperativsystemenes tilgang til maskinvaren.



Figur 14: Plattformvirtualisering

Det er i dag svært mange leverandører av løsninger til virtuelle miljøer. For å nevne noen av de største, har vi blant annet VMware[68], VirtualBox[69] fra Sun Microsystems,

Virtual PC[70] fra Microsoft, Xen[71] fra XenSource og Kernel-based Virtual Machine[72] fra Qumranet.

Et av kriteriene for å kunne benytte en virtuell løsning ved testing av exploits, er muligheten til å kunne tilbakestille testmiljøet til en sikker tilstand. Dette må gjøres før en test skal kjøres, og vil sikre at vi får de samme testkriteriene hver gang.

Virtualisering har i den senere tid blitt utbredt for testing av malware. Det negative med dette er at det også er noe som skapere av malware har fått med seg. Dette har resultert i at noen typer malware er laget for å skjule seg, eller oppføre seg anderledes når det er i et virtuelt miljø. En metode dette kan gjøres på, er for eksempel ved å bruke Red Pill[73] som automatisk detekterer om man er i et virtuelt miljø eller ikke. Det finnes også andre måter å detektere virtuelle miljøer, samt måter å forsvare seg mot disse[74]. Det er viktig å være obs på at slike metoder finnes, og at man må ta høyde for dette hvis man kvalitetssikrer tvilsomme exploits i et virtuelt miljø.

4.5.2 Sikring av testmiljø

Det er viktig at testmiljøet er tilstrekkelig sikret ved testing av exploits. Hvis man tester en exploit som inneholder malware må man sørge for at denne ikke har mulighet til å påvirke noe utenfor testmiljøet.

En mulighet er å benytte virtualisering for sikring. På denne måten vil alle programmer kjøres i et virtuelt miljø og ikke "se" underliggende systemer. Det skal i teorien ikke være mulig å bryte seg ut av et virtuelt miljø, men det kan alltids forekomme feil i programmet som svekker sikkerheten. Hvis man benytter fysiske maskiner, må man sannsynligvis slette og installere alt på harddisken for å kunne være helt sikker på at man får et "friskt" miljø foran hver test.

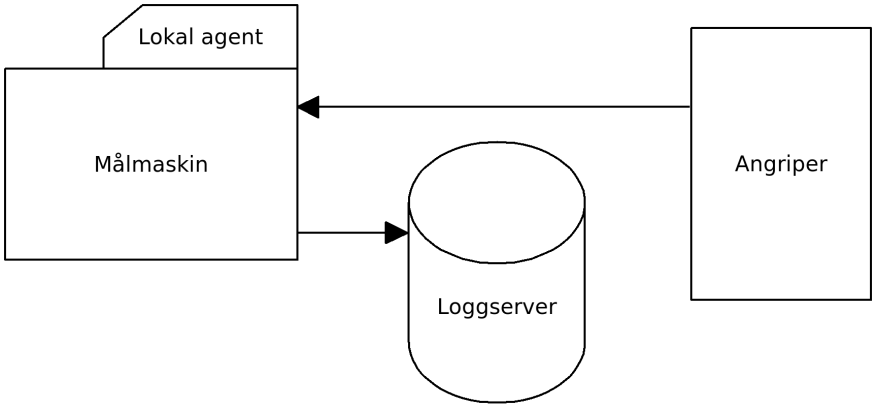
Når det kommer til nettverkssikkerhet har man flere muligheter. Noen virtualiseringsløsninger, som for eksempel Vmware, har bygget inn funksjonalitet så du kan opprette er virtuelt nettverk og bestemme hva den virtulle maskinen kan aksessere på nettverket. Hvis man derimot bruker fysiske maskiner, kan man benytte brannmurer som filterer trafikken eller nettverksutstyr med støtte for VLAN[20]. Man har også alternativet med å fysisk skille testmiljøet fra nettverket. Det sistnevnte er et godt alternativ hvis man ikke trenger kontakt med nettverket.

Fra et brukers perspektiv bør det utarbeides en brukerpolicy hvor man beskriver hvordan testingen skal utføres, og hvordan testmiljøet brukes riktig slik at man kan stole på resultatene.

[20] Virtuelt LAN - en teknikk for å dele et fysisk nettverk opp i flere logiske

4.6 Oppsett av testmiljø

I dette kapittelet vil det vises en oversikt over hvordan testmiljøet kan settes opp sammen med påfølgende forklaring. Se figur 15 for en illustrasjon av oppsettet. Vil vi gi en oversikt over hvilke enheter som trenges og hvordan disse jobber sammen. Det vil komme et forslag til en mer konkret implementasjon i kapittel 5.



Figur 15: Oversikt testmiljø

Angriper
Et angrep kan initieres på to forskjellige måter: via nettverket, eller lokalt på målmaskinen.

Målsystem
Målsystemet er systemet som alle angrepene er rettet mot. Her startes det også en lokal agent før angrepet initieres.

Lokal agent
Som figuren viser er det en agent som sørger for den dynamiske overvåkningen av målsystemet. Denne startes før angrepet blir eksekvert og overvåker filer, prosesser, register og nettverk. Under angrepet kan loggingen enten sendes fortløpende til loggserveren, eller ved endt test.

Loggserver
Loggserveren tar imot loggen som sendes, og starter analyse av disse etter endt eksperiment. I tillegg til å være loggserver er denne maskinen også en webserver, som fortløpende presenterer resultatet til brukerene. Det er også mulig å gi vertsmaskinen rollen som loggserver for å lage et mindre komplisert testmiljø.

4.7 Automatisering

Som nevnt i kapittel 3.5 er det ønskelig med et automatisert system. Dette er blant annet for å gjøre systemet enklest mulig å bruke og unngå at brukere må sitte og manuelt gjøre eksperimentet. Med automatisering vil man også unngå menneskelige feil som ofte kan skje ved repeterende oppgaver.

Når systemet skal automatiseres er det viktig at alle verktøyene som blir brukt har mulighet for å scriptes/programmeres slik at de kan settes sammen med andre verktøy. Dette gjøres enklest ved hjelp av kommando linje, men kan også gjøres ved hjelp av konfigurasjonsfiler.

En fase som det er viktige å automatisere, er oppstart av systemet. Selve eksperimentet kan også automatiseres hvis exploiten har støtte for kommandolinje. Dette må da legges inn etter at testmiljøet er klart. Når enten exploiten er ferdig automatisk, eller hvis det er en manuell exploit og brukeren velger at den er ferdig, er det ønskelig at dataene som ble samlet inn blir overført og importert til databasen som brukes sammen med presentasjon automatisk. Ut fra databasen og mot presentasjon kan det legges på filtre som filtrerer dataene før de presenteres.

Automatisering av selve testmiljøet innebærer blant annet tilbakestilling av virtuelle maskiner til en sikker tilstand, starting og stopping av virtuelle maskiner. For å automatisere tilbakestillingen, starting og stopping kreves det at det virtuelle miljøet har funksjoner for dette. VMware har kommandoen vmrun som kan benyttes for å styre de virtuelle maskinene. VMware kan også automatiseres ved hjelp av API-scripting[75].

For å automatisere loggingen kreves det av det er mulig og starte/stoppe loggagentene automatisk. På målmaskinen betyr dette at agenten må startes fra vertsoperativsystemet i det virtuelle miljøet, eller en annen maskin i det virtuelle miljøet. PSexec fra Sysinternals kan brukes til å eksekvere filer fra en remote-maskin. For at PCexec skal kunne kjøres mot en Windows XP må ”Simple fileshare” i ”Explorer” under valget”Tools” og ”Folder options” slås av. PSexec kan så kjøres fra den eksterne hosten ved å benytte ”psexec \\winxp-ip -u user -p password -c program.exe”. PSexec eksekverer så programmet på maskinen. Overføring av loggen kan så foregå med foreksempel netcat eller cryptcat tilbake til vertsoperativsystemet.

Dette kan også gjøres med VMware sin vmrun kommando. VMrun kan delvis styre gjesteoperativsystemer med kommandoer som kan kopiere inn/ut filer, starte programmer og så lignende. Dette er nyttig med tanke på å automatisere et angrep eller en agent som skal kjøre på gjesteoperativsystemet. Eksempel på bruk av kommando for å starte program på gjesteoperativsystem: ”vmrun -gu guestosuser -gp guestospassword runProgramInGuest "[standard] WinXP/WinXP.vmxC:\program.exe"”. Overføring av filer eller logg kan gjøres ved hjelp av ”copyFileFromGuestToHost”-kommandoen.

Når loggdataene er overført til vertsoperativsystemet kan resultatene fra testen presenteres på en webside. Denne kan lages med forskjellige språk som for eksempel PHP[76] eller

ASP[77] sammen med SQL spørringer. Ved å bruke en database, foreksempel SQLite[78], MySQL[79] eller PostgreSQL[80] vil dette gjøre det raskt å navigere og filtrere data etter gitte kriterier.

4.8 Policy og brukerveiledning

Brukerveiledningen beskriver hvordan man installerer systemet for kvalitetssikring. Det viser også grunnleggende bruk av systemet og beskriver hvordan man kjører en test.

For å benytte systemet er det viktig og ha en policy med regler for bruk av rammeverket. Det er viktig å følge policyen for og kunne stole på resultatene fra testene som gjennomføres.

Se appendix J og I for eksempler/forslag for hvordan dette kan gjøres.

5 Testmiljø og eksperiment

I dette kapitlet skal vi først beskrive hvordan vi har valgt å sette opp testmiljøet, og hvordan vi løst oppgaven med automatisering og kvalitetssikring. Videre vil det gjennomføres testing av dette miljøet mot tre forskjellige typer eksperiment.

Det første eksperimentet vil bestå av en fiktiv "exploit", der vi selv har laget et script som genererer ulik aktivitet mot filer, prosesser, register og nettverk. Dette vil i hovedsak fungere som en kvalitetssikring av testmiljøet og oppsettet for å bekrefte at vi detekterer den aktiviteten vi er ute etter.

Når den første testen er gjennomført, vil vi gå videre og teste rammeverket på to ulike typer kjente exploits som kan være ønskelig å benytte i penetrasjonstesting. Dette vil også være med å verifisere at testmiljøet fungerer, men på en mer reell måte. Med kjente exploits menes det at de er dokumentert, og at handlinger og endringer på ressursene de påvirker er blitt dokumentert.

Den første av de to reelle exploitene utnytter en sårbarhet i Windows og returnerer et shell tilbake til angriperen. I det siste eksperimentet blir det testet en mer kompleks exploit som gjør flere endringer på målsystemet. Dette vil være den mest utfordrende testen, da det blir generert mye data som må håndteres og filtreres ut under presentasjonen av resultatene.

5.1 Testmiljø

I dette kapitlet vil det komme en oversikt over hvordan vi har valgt å sette opp og konfigurere testmiljøet, i tillegg til at verktøyene vi har tatt i bruk blir presentert. For en oversikt over oppsett av testmiljø, vises det til kapittel 4.5.

5.1.1 Valg av testmiljø

For å kunne sette opp et testmiljø som er identisk med et "vanlig" IT-system, har vi valgt å bruke virtualisering. Bakgrunnen for dette var først og fremst at vi ikke hadde tilgang til egnet utstyr for å sette opp et fysisk testmiljø, samtidig som at et virtuelt miljø både er enklere og raskere med tanke på oppsett av ulike eksperimenter og mulighet for tilbakestilling til sikre/kjente tilstander. I tillegg til dette kan et virtuelt miljø enkelt isoleres fra omverdenen, noe som er høyst nødvendig ved testing av skadelig kode.

For å sette opp dette virtuelle miljøet har vi valgt å benytte VMware Server. VMware Server er gratis og støtter blant annet bruk av "snapshot"-funksjonalitet, som dekker behovet vi har for å kunne rulle systemene tilbake til en sikker tilstand.

5.1.2 Installasjon av testmiljø

Siden valget falt på virtualisering i eksperimentene, var det ønskelig med en maskin som hadde nok minne og CPU til å kjøre flere operativsystemer samtidig uten at det oppstår større flaskehalser. I testmiljøet ble eksperimentene utført på en maskin med 2.6 GHz CPU og 3.2 GB minne. Som baseoperativsystem på vertsmaskinen valgte vi å benytte Linux distribusjonen Debian, siden denne både er fleksibelt og minimalistisk. Poenget er at man må frigjøre så mye som mulig av datamaskinens ressurser, for at gjesteoperativsystemene kan utnyttes maksimalt. I perioden hvor eksperimentet pågikk var den nyeste utgaven av Debian versjon 4.0-r06[21].

Vertsmaskinen ble videre installert med et minimum av forhåndsinstallerte pakker for å ikke beslaglegge mer ressurser enn nødvendig. Pakkene som var en absolutt nødvendighet for at WMware Server og miljøet skulle fungere var "build-essentials", "linux-headers-2.6.18-6-686", "openSSH" og "Apache2". Valget av VMware Server ble den nyeste utgaven, versjon 2[68]. Denne versjonen skiller seg ut ved at den har gått vekk fra det tradisjonelle "VMware Server console"-grensesnittet, og heller har valgt å benytte webgrensesnitt. Dette vil si at man gjør all konfigurasjon (opprettelse, fjerning, starting og stopping av maskiner) fra nettleseren[22].

Videre konfigurering av VMware gikk ut på å skille testmiljøet fra resten av nettverket. Av prinsipp så vi dette som en nødvendighet for å ivareta sikkerheten. Dette ble gjennomført ved å sette begge de virtuelle maskinene i det pre-konfigurerte virtuelle nettverket, vmnet0. Dette nettverket er som standard konfigurert som "host-only", som tilsier at gjestemaskinene kun har lov å snakke med hverandre og vertsmaskinen.

Som beskrevet i kapittel 4.2.1 om overflateanalyse, trengs det en metode for å utføre analyser/tester av filene til maskinen som skal overvåkes før og etter test. For å utføre dette har vi valgt å benytte det tidligere diskuterte programmet VMware-mount.pl, som monterer filsystemet til gjesteoperativsystemet direkte på vertsmaskinen. For å få dette programmet til og fungere trengs de to bibliotekene *libfuse2* og *libfuse*, samtidig som man må laste modulen fuse ved hjelp av kommandoen "modprobe fuse".

Når vertsmaskinen var satt opp fortsatte vi med gjestemaskinene (Målmaskin og angriper), som kun var standardinstallasjoner av Windows XP professional med Service Pack 2. Logging og analyse ble gjort direkte på vertsmaskinen.

5.1.3 Valg av verktøy for overvåking

Etter å ha testet forskjellige verktøy til de forskjellige ressursene, valgte vi ut de vi mente var mest egnede til overvåkingen. Disse er beskrevet nedenfor med kryssrefereranser til

[21] Per dags dato er versjon 5 (kodenavn "Lenny") sluppet ut
[22] https://server-ip:8333

datainnsamling i kapittel 4. Ved å benytte de utvalgte verktøyene, mener vi at de tidligere nevnte ressursene blir overvåket på en tilfredsstillende måte.

I overflateanalysen har vi valgt å bruke flere ulike verktøy til å sammenligne de forskjellige tilstandene til målmaskinen. For å lage fingeravtrykk av filene ble det brukt Afick og dens implementasjon av MD5 (se kapittel 4.2.3), mens for å hente ut filenes metadata ble det tatt i bruk TCT sin mactimes og mac-robber. I kapittel 4.2.3 ble det nevnt at en av fordelene til forensicsverktøyet, Afick, var dens muligheter for blant annet å kombinere MD5 og mactimes. Grunnet mactimes sin enkle fremvisning av resultater(tidslinjen), så ble dette verktøyet sett på som enklere å benytte en Afick i forbindelse med analysedelen.

For å detektere endringer i antall kjørende/avsluttende prosesser, gjennom analyse av minnedumper, ble verktøyet Volatility tatt i bruk med henholdsvis modulene psscan og pslist.

Når overflateanalysen ikke lenger strekker til, eller man ønsker mer detaljerte data, så ble den dynamiske analysen tatt i bruk (se kapittel 3.2). Her ble det i eksperimentdelen stort sett brukt Sysinternals sin Process Monitor, Procmon. Denne ble konfigurert til å hooke alle systemkall som omhandlet filer (leste, skrevne, opprettede og slettede), prosesser (opprettede, startede og avsluttede) og register (leste, skrevne, opprettede og slettede). En stor fordel med dette programmet var at vi kunne se hvilken prosess som utførte de ulike handlingene.

En annen ressurs som ble overvåket dynamisk var nettverk. Her valgte vi å benytte VMware sin mulighet til å sniffe VMnet-grensesnittene (kapittel 4.2.5), direkte fra vertsmaskinen. Dette gjør at man unngår å benytte en egen agent på målmaskinen, noe som forsikrer et pålitelig resultat. Selve sniffingen ble gjort med TCPdump.

For automatiseringen sin del ble det benyttet VMrun til stort sett alle delene som omhandlet start, stopp, tilbakestilling, filkopiering og eksekvering. Dette for å unngå mest mulig selvlaget aktivitet på målmaskinen.

De ulike verktøyene vi har tatt i bruk har gode scriptingmuligheter, slik at vi har fått muligheten til å kunne automatisere testmiljøet i stor grad.

5.2 Automatisering av testmiljø

For å kunne benyttes oss av fordelene nevn i kapittel 3.5 og 4.7 setter vi opp automatisering for enklere å kunne gjennomføre testene. Dette setter vi opp ved hjelp av scripting av valgte verktøy, i tillegg til VMware og VMrun.

VMrun kan gi oss oversikt over kjørende VMware-maskiner, samtidig kan vi kontrollere de med blant annet restart og ved tilbakestilling av et snapshot til en sikker tilstand.

Vi kan starte og stoppe agenten på målmaskinen ved hjelp av vmrun og funksjonen "runProgramInGuest". For å starte agenten kjøres agent-scriptet (appendix B) på målmaskinen ved hjelp av vmrun-kommandoen:

```
''vmrun -h https://127.0.0.1:8333/sdk -u user -p password -gu guestosuser -gp guestospassword
     runProgramInGuest "[standard] Victim\_WinXP/Victim\_WinXP.vmx" "C:\agent\start.bat"''
```

Programmet stop.bat stopper scriptet. Loggen kan så overføres fra målmaskinen ved hjelp av vmrun og "copyFileFromGuestToHost"-kommandoen. Selv om det i kapittel 4.3.5 ble diskutert ulemper ved kopiering av loggfiler til analyse-maskin *etter* utført test, har vi valgt å benytte denne metoden på grunnlag av begrensning av tid og med tanke på at dette kun er et "proof-of-concept".

Agenten for å starte logging av filaktivitet, prosesser og register ble gjort på følgende måte på målmaskinen:

```
::Starter process monitor minimert og logger til fil
start C:\agent\procmon.exe /AcceptEula /NoFilter /quiet /minimized /backingfile log.pml

::Vent til procmon er klar
C:\agent\procmon.exe /AcceptEula /WaitForIdle
```

For automatiserings skyld er det viktig å bruke "/AcceptEula"-parameteren for at ikke Process Monitor skal stoppe opp på lisensavtalen. "/NoFilter" slår av Process Monitor sine standardfiltre, og gjør at vi heller kan filtrere dataene selv i analysefasen. "/WaitForIdle" er viktig å ta med slik at ikke den fiktive exploiten starter å kjøre før Process Monitor er klar til og begynne å logge. Hvis ikke dette er klart kan vi miste data.

Logging av nettverk ble gjort ved å overvåke det virtuelle nettverket målmaskinen var på. På denne måten vil vertsmaskinen oppføre seg som en gateway, og all aktivitet vil bli fanget opp. TCP dump ble kjørt på loggserveren med følgende parametre: "tcpdump -i vmnet1 -A -tttt -s0 -w dump.dmp"

Når testen skal gjennomføres trengs det kun å starte "autoscript.sh" (appendix A) på vertsmaskinen. Dette sørger for at målmaskinen blir tilbakestilt til en sikker tilstand, og vet på denne måten at systemet befinner seg i en kjent tilstand. Videre startes loggagenten slik at angrepet kan utføres. Systemet venter så på et tastetrykk fra brukeren i det eksperimentet skal avsluttes. I mellomtiden kan angrepet kjøres på målmaskinen.

For å tilbakestille målmaskinen brukes "revertToSnapshot"

```
...
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox revertToSnapshot "[standard] Victim_WinXP/Victim_WinXP
     .vmx" victim-safe
...
```

Når systemet er klart står det og logger aktivitet på ressursene, og angrepet kan kjøres.

Når angrepet er ferdig trykker man en tast. Agenten stoppes så med VMrun, og loggfilen som agenten genererer overføres til vertsmaskinen for videre analyse.

```
...
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp test
    runProgramInGuest "[standard] Victim_WinXP/Victim_WinXP.vmx" "C:\agent
    \stop.bat"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp test
    copyFileFromGuestToHost "[standard] Victim_WinXP/Victim_WinXP.vmx" "C
    :\agent\log.csv" /home/test/log.csv
...
```

Agenten avsluttes og konverterer loggfilen til CSV-format:

```
::Avslutter process monitor og lagrer . pml log
C:\agent\procmon.exe /AcceptEula /terminate
::Aapner process monitor log og konverterer den til CSV format
C:\agent\procmon.exe /AcceptEula /NoFilter /OpenLog log.pml /SaveAs log.
    csv
```

På loggserveren avsluttes tcpdump-loggingen og begge loggene overføres til host-maskinen for å bli analysert. På hostmaskinen startes et program som står for selve analysen.

5.3 Analysemotor og webside

Det var vanskelig å finne en løsning som hadde ønsket virkemåte, så vi valgte derfor å utvikle et eget verktøy. Løsningen ble utviklet i PHP[76] og bruker relasjonsdatabasen SQLite[78] for lagring av data. I grove trekk består løsningen av to deler:

- **Analyse**
  Et program som henter ut relevant data fra loggfilene, konverterer dem til et felles format og lagrer disse i en database. Det er dette programmet som inneholder filteret som kan gi poeng ut i fra om det er en antatt falsk positiv eller en antatt risiko. Du kan lese mer om dette i kapittel 4.3.
- **Presentasjon**
  En webløsning som presenterer resultatene for brukeren basert på dataene funnet i analysedelen. Dette står det mer om i kapittel 4.4.

Under finner du en utskrift av prosjektets fil-tre. Her er det verdt å legge merke til at alle filene for webløsningen er i katalogen "shared/", men resten av filene er relevant for analysen.

```
kse                                          # Prosjektets rot
|
|-- datastore
|   |-- sqlite-factory.php                   # Klasse som produserer databasetilkoblinger
|   |-- sqlite-file-event-dao.class.php      # Disse fire klassene staar for Lagring/henting
|   |-- sqlite-ip-network-event-dao.class.php  # av hendelser for henholdsvis filer, nettverk,
|   |-- sqlite-process-event-dao.class.php   # prosesser og register fra databasen
|   `-- sqlite-register-event-dao.class.php
|
|-- dto                                      # DTO (Data Transfer Object)-klasser
|   |-- file-event.class.php
|   |-- ip-network-event.class.php
|   |-- process-event.class.php
|   `-- register-event.class.php
|
|-- filters                          # Filter som gir en hendelse "falske positiv"
|   `-- insert-filter.php            # og "risiko"-poeng basert paa gitte regler
|
|-- lese-fra-afick.php               # Lese og konvertere afick-logger til felles format
|-- lese-fra-macrobber.php           # Lese og konvertere macrobber-logger til felles format
|-- lese-fra-procmon.php             # Lese og konvertere procmon-logger til felles format
|-- lese-fra-tcpdump.php             # Lese og konvertere tcpdump-logger til felles format
|
|-- logs                             # Katalog hvor programmet henter loggfiler
|   |-- afick-logger
|   |   `-- afick.txt
|   |-- macrobber-logger
|   |   `-- mactime.txt
|   |-- procmon-logger
|   |   `-- log.csv
|   `-- tcpdump-logger
|       `-- log.txt
|
|-- main.php                         # Fil for aa starte analyse-programmet
|
|-- scripts
|   `-- empty-db.sh                  # Script for aa opprette en tom database
|
`-- shared                           # Katalogen som inneholder webloesningen
    |-- activity.php                 # Program som gir aktivitetsgrafen data
    |
    |-- amstock                      # Diverse filer for grafen
    |   |-- amstock.swf
    |   |-- amstock_settings.xml
    |   `-- swfobject.js
    |
    |-- img                          # Bilder benyttet av webloesningen
    |   |-- akt-o-meter.png
    |   |-- evil-o-meter.png
    |   `-- pil.gif
    |
    |-- index.php                    # Indeksside for webloesningen
    |
    |-- list.php                     # Opplisting av hendelser (underlink)
    |
    `-- summary.php                  # Forside i webloesningen
```

Det er også lagt ved løsningens SQL-definisjoner, TO-klasser[81], og startfilen i vedlegg E.

5.4 Eksperiment

Av hensyn til kvalitetssikringens troverdighet så vil eksperimentet bestå av tre deler: en fiktiv "exploit", en kjent exploit og en mer kompleks exploit. Dette gjøres for å kvalitetssikre miljøet og se at de relevante endringene blir registrert.

Siden det raskt blir mye støy når operativsystemet kjører, har vi valgt å kjøre eksperi-

mentene over så kort tid som mulig for å unngå dette. Vi ser at det hadde vært nyttig å logge over tid for eksempelvis å detektere logiske bomber. Siden vi ikke har noe optimalt filter for å fjerne støy, er det vanskelig å få noe nyttig ut av å logge over tid, da eventuelle funn vil drukne i mengden av data.

Hvert av eksperimentene vil kjøres flere ganger, dette for å være sikker på at man får det samme resultatet hver gang.

5.4.1 Fiktiv exploit

Dette deleksperimentet går ut på å kjøre et "angrep" som gjør kjente handlinger for gitte ressurser. Vi benytter en fiktiv "exploit" som har blitt laget for å gjøre mindre endringer som påvirker ressursene vi overvåker. Dette gjøres for å verifisere at rammeverket detekterer endringene.

Ressursene som blir påvirket av den fiktive exploiten er filer, prosesser, nettverk og register. Minnet blir også påvirket av disse handlingene, men vil ikke bli vurdert i resultatdelen på grunn av omfanget, og at vi har begrensning på tid.

Eksperiment 1 - Test av fiktiv exploit

Type exploit	fiktiv exploit (Egenprodusert script)
Deltagere	KSE-prosjektgruppen
Dato	04.05.2009

Formål
Teste automatisering og loggsystem for kvalitetssikring av exploit, og for å kvalitetssikre selve systemet ved å se at det faktisk registrerer de endringene vi påfører systemet. Dette gjøres ved kjøring av vår "fiktive exploit", et batchscript.

Forventet resultat
Detektere endringene batchscriptet gjør. Bevisste handlinger er:

- Starte calc.exe fra C:\Windows\System32
- Avslutte Notepad.exe med tskill.exe
- Lage den nye filen "C:\windows\system32\fiktiv.txt" med innholdet "Fiktiv"
- Lese filen C:\fil1.txt
- Endre på filen C:\fil2.txt
- Slette filen C:\fil3.txt
- Laste ned filen http://hovedprosjekter.hig.no/v2009/imt/is/kse/data/themes/kse/img/kse-logo.png ved hjelp av wget[23] for windows
- Legge til "Keyname" under "HKLM\Software\TestCompany" i registeret

[23] Wget er et program som henter innhold fra internettsider

- Lese ut ”HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce” fra registeret
- Endre ”HKCU\textbackslash Control Panel\Desktop\PaintDesktopVersion” fra 0 til 1
- Slette ”HKLM\Software\TestCompany\User”-nøkkelen fra registeret

Utstyr og miljø
Testen ble utført i Windows XP SP2, og kjørt på VMware Server 2. Før den fiktive exploiten kjøres må det opprettes tre filer, startes en prosess og legges til en registernøkkel. Dette er for å ha data som kan endres når den fiktive exploiten starter. Filene C:\fil1.txt, C:\fil2.txt og C:\fil3.txt må eksistere. Register nøkkelen ”HKLM\Software\TestCompany\User er opprettet og C:\Windows\system32\notepad.exe kjører.

Dette ble gjort med følgende batch-script:

```
:: PRE: Filer, registernoekkel og prosess som finnes foer start
echo fil1.txt > C:\fil1.txt
echo fil2.txt > C:\fil2.txt
echo fil3.txt > C:\fil3.txt
REG ADD HKLM\Software\TestCompany /v User /t REG_SZ /d best /f
start notepad.exe
```

Filter som ble brukt for å fjerne falske positiver var skrudd på. Aktivitet fra Procmon.exe og VMware tools ble filtrert bort.

Prosedyre

1. Automatisert klargjøring av systemet ved hjelp av autoscript (appendix A)
2. Laste opp og starte fiktiv.bat (appendix C) ved hjelp av VMrun.
   - vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp test copyFileFromHostToGuest "[standard] Victim_WinXP/Victim_WinXP.vmx" /svn/trunk/fiktiv/fiktiv.bat ”C:\fiktiv.bat”
   - vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp test runProgramInGuest "[standard] Victim_WinXP/Victim_WinXP.vmxC:\fiktiv.bat"
3. Følgende bat-fil ble opprettet for å simulere en exploit:

```
::Starter prosess
::Starte prosess
start C:\Windows\system32\calc.exe

::Avslutter prosess
C:\Windows\system32\tskill notepad

::Skrive til ny, aapne, endre og slette fil
echo Fiktiv > C:\windows\system32\fiktiv.txt
type C:\fil1.txt
echo endring >> C:\fil2.txt
del C:\fil3.txt

::Laste ned fil fra internett
C:\wget.exe http://hovedprosjekter.hig.no/v2009/imt/is/kse/data/themes/kse/img/kse-logo.png

::Skriv til, lese fra, endre og slette register
C:\Windows\system32\REG ADD HKLM\Software\TestCompany /v Keyname /t REG_SZ /d 111-22 /f
C:\Windows\system32\REG QUERY "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce" /s
C:\Windows\system32\REG ADD "HKCU\Control Panel\Desktop" /v PaintDesktopVersion /t REG_DWORD /d 0
     x00000001 /f
C:\Windows\system32\REG DELETE HKLM\Software\TestCompany /v User /f
```

Observasjoner
For ressursen "filer", detekterer både overflateanalyse og dynamisk analyse denne aktiviteten. Afick rapporterer blant annet disse endringene:

```
...
new unknown_type :  /mnt/victim_XP/WINDOWS/system32/fiktiv.txt
deleted unknown_type :  /mnt/victim_XP/fil3.txt
changed unknown_type :  /mnt/victim_XP/fil2.txt
...
```

Mactimes finner blant annet følgende informasjon:

```
...
Thu May 07 2009 14:41:11   21 mac -r-------- 0   0   7218   /mnt/victim_XP/fil2.txt
Thu May 07 2009 14:41:11   11 .a. -r-------- 0   0   6648   /mnt/victim_XP/fil1.txt
...
```

Vi ser her at "fil2.txt" har blitt modifisert, aksessert og endret, noe som stemmer med hvordan Windows oppdaterer MAC-times. "fil1.txt" ser vi at kun har blitt aksessert.

Slettingen ble detektert av den dynamiske loggen:

```
2009-05-07 11:45:13 C:\fil3.txt cmd.exe
```

Det samme ble skrivingen til fil:

```
2009-05-07 11:45:13 C:\fil2.txt cmd.exe
2009-05-07 11:45:13 C:\WINDOWS\system32\fiktiv.txt  cmd.exe
```

Legge til nøkkel i register:

```
Dato                Startprogram    Noekkel

2009-05-07 11:45:20 reg.exe         HKLM\Software\TestCompany\Keyname
```

Lesing fra register:

```
Dato                Startprogram    Noekkel

2009-05-07 11:45:21 reg.exe         HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
```

Endring i register:

```
Dato                Startprogram    Noekkel                                          Verdi

2009-05-07 11:45:22 reg.exe         HKCU\Control Panel\Desktop\PaintDesktopVersion  1
```

Sletting fra register:

```
Dato                Startprogram    Noekkel

2009-05-07 11:45:23 reg.exe         HKLM\Software\TestCompany\User
```

Nettverkstilkoblingsforsøket ble også detektert.

Konklusjon
Vi finner alle endringene vi har påført systemet.

Selv om vi fant det vi skulle, ser vi at selv et kort, kjent eksperiment lager mye støy. Dynamisk ble det lest 799 filer og 1300 registernøkler. Det ble opprettet 280 filer og 106 registernøkler.

Dette er en kontrast til de resterende resultatene som viser klart mindre støy. For eksempel er det kun 47 skrivinger i registeret.

Som nevnt tidligere ble eksperimentet utført med filter for falske positiver. Listingen nedenfor viser resultatet av overvåkningen av fiktiv exploit.

Listing 13: Oversikt resultat fiktiv exploit

```
            Opprettet  Startet  Avsluttet
Prosesser   10         9        8

            Lest  Opprettet  Skrevet  Slettet
Filer        799  280        4        1
Register    1300  106        47       1

Nettverkstilkoblinger opprettet: 3
```

Det burde altså implementeres bedre filtre for lesing og oppretting av filer og register, da vi kun opprettet to filer (fiktiv scriptet og fiktiv.txt) og leste ni filer (prosesser og en tekstfil). Deler av lesingen kommer fra prosesser som blir startet (lest fra fil og inn i minnet), mye av aktiviteten kan derfor ses på som støy.

5.4.2 Kjent exploit

I dette eksperimentet vil vi teste "den kjente exploiten". Dette vil gi en mer reell test, samtidig som vi har god kontroll på hva som skjer da exploiten er godt dokumentert.

Eksperiment 2 - Test av kjent exploit

Type exploit:	Metasploit shell ms08_067_netapi
Deltagere	KSE-prosjektgruppen
Dato	08.05.2009

Formål

Teste en reell exploit for å verifisere at systemet fungerer og at vi klarer å detektere endringer.

Exploiten vi tester ligger i Metasploit-rammeverket og kalles ms08_067_netapi. Denne utnytter en sårbarhet i Windows "Server"-service[82] som gjør det mulig å eksekvere egen kode over et nettverk.

Forventet resultat

- Det vil bli generert nettverkstrafikk mellom angriper og offer på porter relatert til Netbios (137,138,139 eller 445)
  - Utveksling av informasjon om operativsystemversjon (angriper forsøker å identifisere offer)
  - Angriper prøver å logge seg på et "IPC$"-share på offer

- Det lyttes på innkommende nettverkskoblinger på port 4444
- Nettverkstrafikk mellom angriper og offer på port 4444 (porten som offeret lytter på etter kompromitering)
  - Tekst som blir skrevet til skjerm i opprettet shell (cmd.exe)
- Filen "cmd.exe" blir lest av tjenesten som har ansvar for fildeling i Windows
- Prosessen "cmd.exe" blir opprettet av tjenesten som har ansvar for fildeling i Windows

Utstyr og miljø

Testen ble utført i Windows XP SP2, kjørt på VMware Server 2. Windows brannmur var slått av.

Filter som ble brukt for å fjerne falske positiver var skrudd på. Aktivitet fra Procmon.exe og VMware tools ble filtrert bort.

Prosedyre

1. Automatisert klargjøring av systemet ved hjelp av autoscript (appendix A)
2. Startet angrepet fra angriper-maskinen med Metasploit
   - Valgte automatisk oppdagelse av operativsystem
   - Valgte exploit Windows\smb\ms08_067_netapi
   - Valgte payload bind_shell_tcp
   - Valgte lokal-port 4444
3. Ventet på et suksessfullt resultat, og avsluttet testen ved å fortsette kjøringen av autoscript.

Observasjoner

For ressursen filer detekteres aktiviteten både ved hjelp av overlateanalyse og dynamisk analyse.

Afick rapporterer blant annet disse endringene:

```
...
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/SAM
...
```

I Afick-loggen ovenfor og Mactimes-loggen nedenfor kan man se at "SAM"-filen har blitt endret. Det er ikke forventet at passord-filen skrives under testing av den kjente exploiten, så det er sannsynligvis "støy" fra operativsystemet.

Mactimes finner blant annet følgende informasjon:

```
...
Fri May 08 2009 15:28:27    8192 m.c -r-------- 0  0  3514  /mnt/victim_XP/WINDOWS/system32/config/SAM.LOG
...
Fri May 08 2009 15:30:00  388608 .ac -r-------- 0  0  2241  /mnt/victim_XP/WINDOWS/system32/cmd.exe
...
```

Som nevnt genererer operativsystemet ofte endel støy. Enda et eksempel på dette er at Windows leser mellomlagrede brukerinnstillinger fra registeret. Dette kan du se fra loggen nedenfor.

Skrevne registerverdier (verdi og type er utelatt):

```
Dato                Startprogram    Noekkel

2009-05-08 15:29:09 lsass.exe       HKLM\SAM\SAM\Domains\Account\Users\000003EB\F
2009-05-08 15:29:09 lsass.exe       HKLM\SAM\SAM\Domains\Account\Users\000003EB\F
```

Et utdrag av hvilke filer som ble skrevet:

```
Dato                Operasjon   Filnavn                                  Startprogram

..
2009-05-15 15:28:30 CreateFile  C:\WINDOWS\Prefetch\CMD.EXE-087B4001.pf svchost.exe
2009-05-15 15:28:31 CreateFile  C:\WINDOWS\system32\cmd.exe             svchost.exe
2009-05-15 15:28:31 WriteFile   C:\WINDOWS\Prefetch\CMD.EXE-087B4001.pf svchost.exe
..
```

Følgende prosesser ble opprettet:

```
Dato                Startprogram                            Filnavn

2009-05-15 15:29:09 C:\agent\Procmon.exe                    cmd.exe
2009-05-15 15:29:09 C:\WINDOWS\system32\cmd.exe             VMwareService.exe
2009-05-15 15:28:47 C:\WINDOWS\System32\cmd.exe             svchost.exe
2009-05-15 15:28:30 C:\WINDOWS\system32\wbem\wmiprvse.exe   svchost.exe
```

Nettverkstilkoblinger mellom angriper og målmaskin ble detektert. Nedenfor vises et utdrag av nettverksaktiviteten:

```
...
2009-05-08 15:28:47.813169 IP 192.168.105.128.1155 > 192.168.105.129.microsoft-ds: . ack 5593 win 16733
E..(&B@....;..i...i........m....P.A]ot.. FHEPF
2009-05-08 15:28:48.582322 IP 192.168.105.129.4444 > 192.168.105.128.1157: P 1:40(39) ack 1 win 64240
E..O..@....>..i...i..\...Q.r8.1.P...R...Microsoft Windows XP [Version 5.1.2600]
2009-05-08 15:28:48.773031 IP 192.168.105.128.1157 > 192.168.105.129.4444: . ack 40 win 17481
E..(&C@....:..i...i....\8.1..Q..P.DI.E.. FHEPF
2009-05-08 15:28:48.774128 IP 192.168.105.129.4444 > 192.168.105.128.1157: P 40:105(65) ack 1 win 64240
E..i..@....#..i...i..\...Q..8.1.P.......
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
...
2009-05-08 15:28:45.606468 IP 192.168.105.128.1155 > 192.168.105.129.microsoft-ds: P 662:739(77) ack 480 win
     17041
E..u&.@.......i...i.............P.B........I.SMBu...... ...................L............\\192.168.105.129\
     IPC$.?????.
...
```

Konklusjon

Alle punktene i "Forventet resultat" ble funnet igjen i de innsamlede dataene.

I nettverksloggen ser man blant annet Netbios-trafikk mellom angriper og offer. Etterhvert ser man også endel trafikk mellom angriper og port 4444 på offeret. Dette viser at exploiten har startet en tjeneste som lytter på port 4444 på offeret.

I exploitens dokumentasjon er det beskrevet at den først henter informasjon fra offeret for å finne ut hvilket operativsystem og servicepack som kjøres (Eng.: fingerprinting). I nettverksloggen ser man at offeret sender tekst som "Windows 2000 5.0", derfor er det nærliggende å tro at det er denne informasjonen angriperen baserer seg på.

Teksten "\\192.168.105.129\\IPC$" viser også at det er blitt sendt en forespørsel til "IPC$"-sharet på offeret.

Når angriper koblet til port 4444 på offer, blir det åpnet et "cmd.exe"-shell. Her er det også mulig å finne igjen følgende tekst i loggen:

```
Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft
    Corp.

C:\WINDOWS\system32>
```

Dette samsvarer med teksten som "cmd.exe" skriver til skjermen hver gang det startes.

Hvis man ser på dataene fra Procmon vil man også se at det startes en prosess like før teksten ovenfor dukket opp i nettverksloggen. Dataene viser at "svchost.exe" først leser filen "cmd.exe" og så oppretter en ny prosess.

Følgende statistikk ble totalt detektert, minus filterene:

```
           Lest  Opprettet  Skrevet  Slettet
Filer       398  98          52       0
Register    748  38          19       0

           Opprettet  Startet  Avsluttet
Prosesser  3          3        3

Nettverkstilkoblinger opprettet: 13
```

5.4.3 Kompleks exploit

I dette eksperimentet vil det testes en mer kompleks exploit som gjør flere endringer på målsystemet, og dermed genererer mer data for for resultatet. Dette er en stor utfordring siden vi må passe på at det ikke filtreres bort relevante data som er generert av exploiten.

Eksperimentet er inkludert etter forslag fra ekstern veileder, Ernst & Young. Eksperimentet utnytter ikke direkte en sårbarhet, men måten Windows håndterer autokjørfunksjonen. Funksjonaliteten til denne koden er på mange måter likt forskjellige typer payloads som er inkludert i exploits. Dette ansees som en høyst relevant testcase siden det kan vise deteksjon av ulovlig informasjonsinnsamling på flere nivå.

Eksperiment 3 - Test av kompleks exploit

Type exploit:	Gonzor Switchblade
Deltagere	KSE-prosjektgruppen
Dato	13.05.2009

Formål

Teste en mer avansert kode/exploit enn det foregående eksperimentet. Koden er satt sammen av mange mindre verktøy, og berører store deler av systemet. Fordelen er at det skaper mye fil- og registeraktivitet, som forhåpentligvis vil detekteres av analysen.

Exploiten som testes heter Gonzor Switchblade[83] og er en såkalt USB-slurper. Denne utnytter en svakhet i Windows XP, der man ved emulering av en CD-ROM automatisk kan kjøre innholdet (payloaden) på en USB-lagringsenhet. Denne payloaden vil typisk samle inn informasjon om systemet, som for eksempel Windows lisensnøkler, lagrede passord, nettverkstjenester, etc.

Forventet resultat

Switchblade er et program med svært mye funksjonalitet, så vi har derfor sett oss nødt til å kun ta for oss et utvalg av forventede resultater. Vi har valgt å dra ut funksjonaliteten i listen nedenfor basert på hvor mye potensiell skade det kan gjøre på systemet.

- Eksekvering av flere ulike verktøy for bruk til innsamlingen
- Installasjon av bakdør (VNC)
  - Opprettelse av registernøkler og filer
- Lesing av lagrede passord i Internet Explorer
- Lesing av SAM-databasen
- Lagring av innsamlet data tilbake til lagringsenheten
  - E:\System \logs \
- Kontakte en webtjener for å finne ekstern IP

Utstyr og miljø

Testen ble utført i Windows XP SP2, kjørt på VMware Server 2.

Filter som ble brukt for å fjerne falske positiver var skrudd på. Aktivitet fra Procmon.exe og VMware tools ble filtrert bort.

Prosedyre

1. Automatisert klargjøring av systemet ved hjelp av autoscript (appendix A)
2. Startet det lokale "angrepet" ved å sette inn minnepinnen
3. Monitorering av CPU-bruk for å estimere når angrepet er fullført
4. Ventet på et suksessfullt resultat, og avslutte testen ved å fortsette kjøringen av autoscript.

Observasjoner

Denne exploiten laget en god del mer aktivitet en forventet. Som man kan se fra oppsummeringen (vedlegg H), var det store mengder lesing og skriving til filer og register. Prosess-aktiviteten var også uvanlig høy i forhold til hvor kort tid koden kjørte.

Når det gjelder startede og stoppede prosesser, så ble det registrert 28 opprettede, og 28 terminerte. Prosessene ble eksekvert og terminert i en sekvensiell rekkefølge, noe som indikerer at verktøyene på USB-lagringsenheten ble eksekvert, og avsluttet etter endt oppgave. Antallet startede prosesser samsvarer derimot ikke med antall eksekverbare filer/verktøy som befant seg på lagringsenheten. Videre analyse av resultatet viser at noen standardprogrammer i Windows også ble benyttet i innsamlingsfasen. Dette var blant annet *ipconfig.exe, net.exe, net1.exe, xcopy.exe, sc.exe, attrib.exe* og *reg.exe*. Nedenfor vises en tabell med hva disse ulike prosessene kan benyttes til.

ipconfig.exe	Uthenting av nettverksinformasjon (IP, Gateway, DNS, grensesnitt, etc.)
net.exe/net1.exe	Kontrollering av brukere, grupper, tjenester og nettverksforbindelser
xcopy.exe	Utvidet versjon av det vanlige kopieringsprogrammet, copy.exe
sc.exe	Opprettelse og starting av tjenester
attrib.exe	Endring av read-only attributtet til filer og mapper
reg.exe	Opprettelse, sletting, lesing og modifisering av registernøkler

Når man ser på bruksområdet til disse programmene, er det ikke så vanskelig å skjønne hva de har blitt benyttet til. Ingen systemprosesser ble avsluttet eller startet utenom det vanlige. For en fullstendig liste over startede prosesser, vises det til vedlegg H.1.

I registeret ble totalt 10404 nøkler/verdier lest, 1302 opprettet, 607 modifisert og 15 slettet. I listing 14 er det vist noen eksempler på interessante nøkler som ble opprettet.

Hele listen finnes i vedlegg H.2.

Listing 14: Opprettede registernøkler

```
2009-05-16 11:57:55     services.exe    HKLM\System\CurrentControlSet\Services\WinVNC\Security
2009-05-16 11:57:55     services.exe    HKLM\System\CurrentControlSet\Services\WinVNC
2009-05-16 11:58:06     regedit.exe     HKCU\Software\WinVNC3
2009-05-16 11:58:06     regedit.exe     HKCU\Software\ORL\WinVNC3
2009-05-16 11:58:06     regedit.exe     HKCU\Software\ORL\VNCHooks\Application_Prefs\WinVNC.exe
...
2009-05-16 11:58:13     winvnc.exe      HKLM\Software\ORL\WinVNC3
2009-05-16 11:58:13     winvnc.exe      HKU\.DEFAULT\Software\ORL\VNCHooks\Application_Prefs\winvnc.exe
..
2009-05-16 11:58:18     LaunchPad.exe   HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
2009-05-16 11:58:18     LaunchPad.exe   HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell
     Folders
..
2009-05-16 11:58:45     WIFIKE.EXE      HKCU\Software\NirSoft\WirelessKeyView
```

Her ser man blant annet at prosessen services.exe oppretter en nøkkel for Windows VNC, noe som tyder på at en tjeneste for fjernstyring har blitt lagt til i systemet. Videre ser vi at det er blitt oppført nøkler for denne fjernstyringsprogramvaren flere steder i registeret.

Over 2401 filer ble totalt lest på målmaskinen mens eksperimentet foregikk. Dette tallet kan være litt misvisende, ettersom flere av filene blir lest gjentatte ganger. I listing 15 er det vist noen av de mest suspekte filene som er blitt lest, og hvem som har lest filene. For eksempel er WINvnc et fjernstyringsverktøy som blir installert på systemet, mens FGdump.exe er et verktøy for å hente ut alle passordhasher fra Windows.

Listing 15: Lite utdrag av leste filer på målmaskin

```
2009-05-16 11:57:13     \Device\CdRom1\AUTORUN.INF      svchost.exe
2009-05-16 11:57:20     E:\AUTORUN.INF  Explorer.EXE
2009-05-16 11:57:29     E:\LAUNCHU3.EXE wscript.exe
2009-05-16 11:57:32     E:\SYSTEM\SRC\GO.BAT    wscript.exe
2009-05-16 11:57:33     E:\AUTORUN.INF  LaunchU3.exe
2009-05-16 11:57:34     E:\SYSTEM\SRC\GO.BAT    cmd.exe
2009-05-16 11:57:37     F:\System\Logs\VICTIM\VICTIM-[20090515-115726].log      cmd.exe
2009-05-16 11:57:53     E:\SYSTEM\SRC\VNC\WINVNC.EXE    xcopy.exe
2009-05-16 11:57:53     E:\SYSTEM\SRC\VNC\WINVNC.EXE    System
2009-05-16 11:58:06     E:\SYSTEM\SRC\VNC.REG   regedit.exe
2009-05-16 11:58:12     C:\WINDOWS\WINVNC.EXE   winvnc.exe
2009-05-16 11:58:52     \\127.0.0.1\pipe\svcctl FGDUMP.EXE
2009-05-16 11:59:02     C:\Documents and Settings\test\Local Settings\History\History.IE5\index.dat IEPV.EXE
```

Her ser man blant annet at det er svchost.exe (vertsprosess for dll filer) som starter hele prosessen ved å kjøre AUTORUN.INF fra den emulerte CD-rommen. Etter dette blir både Autorun.inf, LaunchU3.exe og GO.bat lest og kjørt fra USB-lagringsenheten, som videre sørger for at de andre eksekverbare programmene blir igangsatt. Regedit.exe leser SRC\VNC.reg, som indikerer at det er denne filen som blir kjørt før å kjøre spørringer og skrivinger til registeret.

Nettverksaktiviteten var svært liten i dette eksperimentet. Det ble registrert ett navneoppslag, og resten var vanlige Windows-rutiner. I listing 17 kan man se at navneoppslaget var mot ip.gonzor288.com

Listing 16: Opprettede tilkoblinger

```
KildeIp         Kildeport   MottakerIp      MottakerPort
192.168.105.129 netbios-ns  192.168.105.255 netbios-ns
192.168.105.129 1089        239.255.255.250 1900
192.168.105 129 igmp.mcast  net
192.168.105.129 1025        192.168.105.1   domain
192.168.105 1               192.168.105 129
192.168.105.129 proofd      239.255.255.250 1900
192.168.105.129 netbios-dgm 192.168.105.255 netbios-dgm
```

Listing 17: Nettverkspakke i ASCII format

```
2009-05-15 11:57:48.981026 IP 192.168.105.129.1025 > 192.168.105.1.domain:  41501+ A? ip.gonzor228.com. (34)
E..>. ........i...i....5.*..............ip     gonzor228.com.....
2009-05-15 11:57:48.981057 IP 192.168.105.1 > 192.168.105.129: ICMP 192.168.105.1 udp port domain unreachable
     , length 70
E..Z....@.md..i...i...Q.....E..>. ........i...i....5.*..............ip gonzor228.com.....
```

Konklusjon

Alle punktene i "Forventet resultat" ble funnet igjen i de innsamlede dataene. Det eneste punktet det var litt usikkerhet rundt var uthenting av Windows sine passordhasher. Vi ser aktivitet både mot den fysiske SAM-filen og cachede passordhasher i registeret. Grunnen til denne usikkerheten er at vi ser at LSASS.exe er prosessen som utfører alle spørringer mot registeret. I utgangspunktet er LSASS en legitim prosess som brukes til lokal autentisering for brukere og prosesser, men denne kan misbrukes ved hjelp av DLL-injection.

Videre ble det observert at verktøyene på USB-lagringsenheten ble eksekvert etter tur, eksempelvis IEPV.exe for å hente ut passord og netthistorikk fra Internet Explorer, FGdump for å hente ut Windows-passord og Productkey.exe for å hente ut lisensnøkler. Dette samsvarer med kildekoden i "GO.bat", som har ansvaret for å kjøre alle tilleggsverktøyene. Vi observerte også at det ble benyttet verktøy som allerede ligger i operativsystemet.

Ut i fra aktivitet på filsystemet og registeret kunne vi se at Switchblade installerte tjenesten VNCserver, et kjent fjernstyringsverktøy. I tillegg prøvde koden å finne den offisielle IP-adressen, for å kunne gi angriperen muligheten til å fjernstyre målmaskinen ved en senere anledning.

Gjennom hele eksperimentet så vi at det gjentatte ganger ble skrevet til en loggfil på lagringsenheten. Dette er resultatet av alle verktøyene som hentet informasjon fra systemet.

5.5 Oppsummering

Ett enkelt eksperiment ville ikke være nok til å kunne dekke all funksjonaliteten i rammeverket. Det ble derfor utført tre eksperimenter, valgt for å kunne teste rammeverket på forskjellige måter. Dette var henholdsvis:

1. En fiktiv exploit for å først verifisere testmiljøets pålitelighet
2. En kjent exploit for å utføre en reell test
3. En kompleks exploit for å inkludere begge de ovennevnte områdene

Alle eksperimentene ble utført uten problemer, og resultatet ble som forventet. Med dette mener vi at observasjonene var i samsvar med de forventede resultatene. Vi kan imidlertid ikke utelukke hendelser som ikke er observerbare i eksperimentoppsettet.

Det er kun et par ting som er verdt å nevne: I det siste eksperimentet, kompleks exploit, ble det benyttet flere teknikker for å hente ut Windows' passordhasher. Det var derfor litt usikkerhet rundt hvilken teknikker som ga de forskjellige resultatene i loggen. Dette er beskrevet nærmere i eksperiment 3. Ellers har det generelt vært en utfordring å filtrere vekk uønsket data uten å samtidig fjerne noe av relevans. En mulighet er å bygge filter for å fjerne falske positiver, noe det står mer om i kapittel 4.3.4. Vi innser at dette er en problemstilling man ikke kan regne med å finne en komplett løsning på, men som man kontinuerlig må jobbe med for å redusere falske positiver/negativer så mye som mulig.

Vi har lagt vekt på at resultatene skal være etterrettelige, og har derfor kjørt hvert eksperiment flere ganger. Ved å gjøre dette så kan man avdekke eventuelle feil i systemet eller at testsubjektet oppfører seg annerledes for hver gang eksperimentet kjøres. Ved utføring av våre eksperimenter så har vi observert forskjeller i testresultatene, men ved nærmere undersøkelse viser dette seg kun å være ulike typer støy, noe man må forvente i et moderne operativsystem.

6 Diskusjon og konklusjon

Dette kapitlet vil oppsummere prosjektet med de resultater vi har kommet frem til, og refleksjoner rundt disse.

6.1 Diskusjon

Effektmålet for prosjektet har vært å utvikle et hjelpemiddel som EY kan benytte for kvalitetssikring av exploits, og som de senere kan ta i bruk ved penetrasjonstesting ute hos sine kunder. Slik det er nå, kan ikke et generelt rammeverk alene kvalitetssikre exploits, da automatisk behandling av malware alltid vil medføre at enkelte ufarlige operasjoner flagges som mistenkelige (falske positive) og at enkelte uønskede operasjoner ikke oppdages (falske negative). Løsningen vi har kommet frem til gir en god indikasjon på hva en exploit gjør med målsystemet og bør kunne tas i bruk i den innledende fasen for kvalitetssikring. Den vil hjelpe til med å skaffe en oversikt over exploiten, og om den i det hele tatt er egnet til bruk ved penetrasjonstesting.

I det konseptuelle rammeverket har vi lagt vekt på å dokumentere hva som skal finnes på de forskjellige ressursene ved hjelp av verktøyene vi har funnet. Ved å følge disse retningslinjene kan EY supplere testmiljøet med flere verktøy etter ønske og behov. I og med at vi har sett på hva som er viktig å overvåke, kan systemet brukes på andre plattformer ved å benytte verktøy med tilsvarende funksjonalitet.

Etter å ha utført tre eksperimenter, har vi et konsept som virker i praksis, og som kan bli et nyttig verktøy i EYs videre fokus på bruk av exploits ved penetrasjonstesting. Resultatet av de gjennomførte eksperimentene har også stått til våre forventninger.

Ved gjennomføring av eksperimentene finner vi igjen exploitens dokumenterte handlinger ved overvåking av systemet. Dette betyr at testmiljøet fungerer og logger endringene som ønsket.

Vi registrerte også at operativsystemet laget svært mye støy, selv når det kjører på "tomgang". Dette gjelder spesielt ved lesing og oppretting av filer og registernøkler. Mye støy vanskeliggjør filtreringen av dataene. Dette kan medføre at viktige endringer kan "drukne" i all informasjonen. Filtreringen av data bør jobbes videre med for at rammeverket skal være et godt verktøy. Dette ser vi spesielt godt på den siste exploiten, da dette var en kompleks exploit med mye aktivitet.

Siden det ble benyttet både dynamisk og statisk analyse får vi forskjellige innfallsvinkler på overvåkningen. Vi ser at dynamisk analyse er nødvendig, da statisk analyse i hovedsak registrerer de permanente endringene, og mister hendelsesforløpet til exploiten. Likevel ser vi nytten av å verifisere den dynamiske overvåkningen ved å supplere med statisk analyse. Dette er spesielt viktig med tanke på et eventuelt angrep mot testmiljøet og manipulering av resultater.

Vi ser at rammeverket ikke er komplett nok til å brukes alene, men kan brukes som et supplement til eksempelvis binæranalyse. Vi får en god indikasjon på at systemet overvåker de valgte ressursene, men problematikken rundt falske positiver gjør at systemet må utbedres før videre bruk. Systemet bidrar til å skaffe oversikt over endringene som påføres systemet og gir en god indikasjon på hvordan exploiten oppfører seg.

Læringsmålene var å få en dypere forståelse for analyse av exploits i både teori og praksis. Vi har fått god innsikt i analyse av exploits og overvåking av ressurser. Erfaring fra testing i et virtuelt miljø har vært nyttig, da vi har benyttet denne funksjonaliteten i hele prosjektperioden.

Det har vært en svært lærerik prosess å jobbe i et prosjekt med så lang varighet. Vi har tatt i bruk ulike prosjektsstyringsverktøy som har vært til stor nytte for oversikt og fremdrift i prosjektet.

Hvis vi hadde hatt mer tid, ville vi gjerne testet flere exploits. Dette er et spennende, men komplisert område innen sikkerhet, som vi gjerne kunne tenke oss å jobbe mer med.

6.2 Konklusjon

Det har blitt utviklet et rammeverk som tar for seg ulike metoder og verktøy for å kunne overvåke et testmiljø, og presentere resultatet på en oversiktlig måte. Vi har i rammeverket laget forslag til kriterier for å velge ut egnende exploits samt et forslag til et konsept for å behandle falske positiver. Slik rammeverket er i dag, kan vi teste ulike typer exploits og få presentert resultatet på en slik måte at det er mulig å finne igjen forventet aktivitet. Ved videre utvikling har vi stor tro på at konseptet kan bli en viktig del ved kvalitetssikring av exploits i fremtiden.

Alle eksperimentene ble utført uten problemer, og resultatet ble som forventet. Med dette mener vi at observasjonene var i samsvar med de forventede resultatene. Vi kan imidlertid ikke utelukke hendelser som ikke er observerbare i eksperimentoppsettet. Ellers har det generelt vært en utfordring å filtrere vekk uønsket data uten å samtidig fjerne noe av relevans. Vi innser at dette er en problemstilling man ikke kan regne med å finne en komplett løsning på, men som man kontinuerlig må jobbe med for å redusere falske positiver/negativer så mye som mulig.

Det er lagt stor vekt på automatisering og presentasjon for å gjøre løsningen så enkel og effektiv som mulig. Automatiseringen er laget for å ta seg av alle aspekter rundt testen. I praksis betyr dette at kan man kjøre testen kun ved et par tastetrykk. Presentasjonen er i form av en dynamisk webløsning som gir brukeren et overblikk av dataene, samtidig som man har mulighet til å grave seg ned i detaljene. Automatisering og webløsningen er også det vi anser som det største bidraget til fagfeltet.

6.3 Fremtidig arbeid

Fremtidig arbeid bør bestå av å videreutvikle filtreringsfunksjonene og arbeid med de falske positivene. Dette er viktige områder og vil gi løsningen større nytteverdi. Det kan i den sammenheng også være aktuelt å innføre forskjellige regelsett for ulike typer exploits. Applikasjonen er et godt utgangspunkt, men bør videreutvikles for å bli et komplett verktøy. Mer funksjonalitet for eliminering av urelevante resultater og avanserte søkefunksjoner er noe av det vi mener bør utvikles videre.

Et dypdykk innenfor minneanalyse har ikke vært en del av oppgavens omfang, og vi har derfor vektlagt dette mindre enn de andre ressursene. Dette er et komplisert område, og kunne vært en egen oppgave i seg selv. Minneanalyse er derimot et tema som bør tas med i rammeverket for å få en komplett oversikt over alle ressurser og endringer.

Det kan også være ønskelig å utvide verktøykassen med flere verktøy for blant annet andre plattformer. Det kan også være nyttig med mer spesifikke verktøy mot de forskjellige ressursene.

Referanser

[1] Secunia advisories. Accessed on 19/05/2009. [Online]. Available: http://secunia.com/advisories/historic

[2] Securityfocus vulnerabilities. Accessed on 19/05/2009. [Online]. Available: http://www.securityfocus.com/vulnerabilities

[3] milw0rm. Accessed on 19/05/2009. [Online]. Available: http://www.milw0rm.com

[4] L. Haukli, "Analysing Malicious Code: Dynamic Techniques," Master's thesis, NTNU, 2007.

[5] K.-M. Krister, "Automated testing of antivirus coverage," Master's thesis, NTNU, 2008.

[6] C. Willems, T. Holz, and F. Freiling, "Toward Automated Dynamic Malware Analysis Using CWSandbox," 2007.

[7] Metasploit Framework. Accessed on 19/05/2009. [Online]. Available: http://www.metasploit.com

[8] M. Hypponen and H. Malmari, "F-Secure Reports Amount of Malware Grew by 100 percent during 2007," Tech. Rep., December 2007. [Online]. Available: http://www.f-secure.com/en_EMEA/about-us/pressroom/news\/2007\/fs_news_20071204_1_eng.html

[9] D. Turner *et al.*, "Symantec Internet Security Threat Report: Trends for July-December 2007," Tech. Rep., April 2008. [Online]. Available: http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xiii_04-2008.en-us.pdf

[10] Penetration testing guide. Accessed on 19/05/2009. [Online]. Available: http://www.penetration-testing.com

[11] B. Carrier and E. H. Spafford, "Getting physical with the digital investigation process," *International Journal of Digital Evidence*, November 2003. [Online]. Available: https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/archive/2003-29.pdf

[12] A. Årnes, P. Haas, and G. Vigna, "Using a virtual security testbed for digital forensic reconstruction," *Journal in Computer Virology*, Desember 2006.

[13] M. Richmond, "ViSe: The Virtual Security Testbed," Master's thesis, Department of Computer Science, University of California, Santa Barbara, Juni 2005.

[14] P. L. Wedum, "Malware Analysis; A Systematic Approach," Master's thesis, NTNU, 2008.

[15] File System Filter Drivers. Accessed on 19/05/2009. [Online]. Available: http://www.microsoft.com/whdc/driver/filterdrv/default.mspx

[16] FileSystemWatcher Class, .NET API. Accessed on 19/05/2009. [Online]. Available: http://msdn.microsoft.com/en-us/library/system.io.filesystemwatcher.aspx

[17] Y. Kaplan, "API Spying Techniques for Windows 9x, NT and 2000," Tech. Rep., 2000. [Online]. Available: http://www.internals.com/articles/apispy/apispy.htm

[18] IAT Function Hooking. Accessed on 19/05/2009. [Online]. Available: http://sandsprite.com/CodeStuff/IAT_Hooking.html

[19] Hooking Windows NT System Services. Accessed on 19/05/2009. [Online]. Available: http://www.windowsitlibrary.com/Content/356/06/2.html

[20] Kvalitetsikring - ordliste. Accessed on 19/05/2009. [Online]. Available: http://www.kvalitetsikring.no/praktisk/ordliste/

[21] *VMware Disk Mount User's Guide*. [Online]. Available: http://www.vmware.com/pdf/VMwareDiskMount.pdf

[22] Whatis.com, tech dictionary and encyclopedia. Accessed on 19/05/2009. [Online]. Available: http://www.whatis.com

[23] Norman Sandbox. Accessed on 19/05/2009. [Online]. Available: http://www.norman.com/microsites/nsic/

[24] R. Rivest, "The MD5 Message-Digest Algorithm," MIT Laboratory for Computer Science and RSA Data Security, Inc., April 1992, Request for Comments. [Online]. Available: http://tools.ietf.org/html/rfc1321

[25] M. Stevens, A. Lenstra, and B. de Weger, "Vulnerability of software integrity and code signing applications to chosen-prefix collisions for MD5," Tech. Rep., November 2007. [Online]. Available: http://www.win.tue.nl/hashclash/SoftIntCodeSign/

[26] Helix. Accessed on 19/05/2009. [Online]. Available: http://www.e-fense.com/helix/

[27] EnCase Forensic. Accessed on 19/05/2009. [Online]. Available: http://guidancesoftware.com/computer-forensics-ediscovery-software-digital-evidence.htm

[28] AccessData Forensic Toolkit. Accessed on 19/05/2009. [Online]. Available: http://www.accessdata.com/forensictoolkit.html

[29] Timestomp. Accessed on 19/05/2009. [Online]. Available: http://www.forensicswiki.org/wiki/Timestomp/

[30] SetFileDate. Accessed on 19/05/2009. [Online]. Available: http://setfiledate.en.softonic.com/

[31] Immunity: Knowing your'e secure. Accessed on 19/05/2009. [Online]. Available: http://www.immunitysec.com/products-canvas.shtml

[32] Core Security Technologies. Accessed on 19/05/2009. [Online]. Available: http://www.coresecurity.com/

[33] M. Miller and J. Turkulainen. Remote library injection (april 2002). Accessed on 19/05/2009. [Online]. Available: http://www.nologin.org/Downloads/Papers/remote-library-injection.pdf

[34] M. Caceres, "Syscall proxying - simulating remote execution," Tech. Rep., 2002. [Online]. Available: http://www.coresecurity.com/files/attachments/SyscallProxying.pdf

[35] mac-robber. Accessed on 19/05/2009. [Online]. Available: http://www.sleuthkit.org/mac-robber/

[36] mac-times. Accessed on 19/05/2009. [Online]. Available: http://www.sleuthkit.org/

[37] grave-robber. Accessed on 19/05/2009. [Online]. Available: http://www.porcupine.org/forensics/tct.html

[38] Afick. Accessed on 19/05/2009. [Online]. Available: http://afick.sourceforge.net/

[39] Tripwire. Accessed on 19/05/2009. [Online]. Available: http://sourceforge.net/projects/tripwire/

[40] PTfinder - Process and Thread finder (Andreas Schuster). Accessed on 19/05/2009. [Online]. Available: http://computer.forensikblog.de/en/2007/11/ptfinder_0_3_05.html

[41] Ida pro. Accessed on 19/05/2009. [Online]. Available: http://www.hex-rays.com/idapro/

[42] M. Miller, "Metasploit's Meterpreter," Tech. Rep., 2004. [Online]. Available: http://www.metasploit.com/documents/meterpreter.pdf

[43] Process monitoring linux kernel module (jacob bower). Accessed on 19/05/2009. [Online]. Available: jacob.bower@ic.ac.uk

[44] Honeyd Virtual Honeypot. Accessed on 19/05/2009. [Online]. Available: http://www.honeyd.org

[45] libpcap. Accessed on 19/05/2009. [Online]. Available: http://sourceforge.net/projects/libpcap/

[46] Wireshark. Accessed on 19/05/2009. [Online]. Available: http://www.wireshark.org

[47] Ethereal. Accessed on 19/05/2009. [Online]. Available: http://www.ethereal.com

[48] L. W. Wong, "Forensic Analysis of the Windows Registry," Tech. Rep., February 2007. [Online]. Available: http://www.forensicfocus.com/downloads/forensic-analysis-windows-registry.pdf

[49] Windows registry information for advanced users. Accessed on 19/05/2009. [Online]. Available: http://support.microsoft.com/kb/256986

[50] OS Platform Statistics. Accessed on 19/05/2009. [Online]. Available: http://www.w3counter.com/globalstats.php

[51] Regscanner. Accessed on 19/05/2009. [Online]. Available: http://www.nirsoft.net/utils/regscanner.html

[52] Registry hives (windows). Accessed on 19/05/2009. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms724877.aspx

[53] AccessData Registry Quick Find Chart. Accessed on 19/05/2009. [Online]. Available: http://www.accessdata.com/media/en_US/print/papers/wp.Registry_Quick_Find_Chart.en_us.pdf

[54] Autorunsc for windows. Accessed on 19/05/2009. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx

[55] Regmon for windows. Accessed on 19/05/2009. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb896652.aspx

[56] RegFromApp. Accessed on 19/05/2009. [Online]. Available: http://www.nirsoft.net/utils/reg_file_from_application.html

[57] Determining the Last Write Time"of a registry key? Accessed on 19/05/2009. [Online]. Available: http://www.winhelponline.com/articles/12/1/Determining-the-Last-Write-Time-of-a-registry-key.html

[58] Hijackthis. Accessed on 19/05/2009. [Online]. Available: http://www.trendsecure.com/portal/en-US/tools/security_tools/hijackthis/

[59] Dumphive. Accessed on 19/05/2009. [Online]. Available: https://launchpad.net/dumphive

[60] Parse::Win32Registry. Accessed on 19/05/2009. [Online]. Available: http://search.cpan.org/~jmacfarla/Parse-Win32Registry/

[61] Device\PhysicalMemory Object. Accessed on 19/05/2009. [Online]. Available: http://technet.microsoft.com/en-us/library/cc787565.aspx

[62] Generate a memory dump in Windows. Accessed on 19/05/2009. [Online]. Available: http://support.microsoft.com/kb/244139

[63] The Volatility Framework: Volatile memory artifact extraction utility framework. Accessed on 19/05/2009. [Online]. Available: https://www.volatilesystems.com/default/volatility

[64] A. Schuster. Searching for processess and threads in microsoft windows memory dumps. Accessed on 19/05/2009. [Online]. Available: http://www.dfrws.org/2006/proceedings/2-Schuster-pres.pdf

[65] Zgrviewer. Accessed on 19/05/2009. [Online]. Available: http://zvtm.sourceforge.net/zgrviewer.html

[66] Graphviz - Graph Visualization Software. Accessed on 19/05/2009. [Online]. Available: http://www.graphviz.org

[67] MS Windows ASN.1 LSASS.EXE Remote Exploit (MS04-007). Accessed on 19/05/2009. [Online]. Available: http://milw0rm.org/exploits/153

[68] VMware. Accessed on 19/05/2009. [Online]. Available: http://www.vmware.com

[69] VirtualBox (Sun Microsystems). Accessed on 19/05/2009. [Online]. Available: http://www.virtualbox.org/

[70] Virtuap PC (Microsoft). Accessed on 19/05/2009. [Online]. Available: http://www.microsoft.com/virtualpc

[71] Xen (XenSource, Inc). Accessed on 19/05/2009. [Online]. Available: http://www.xen.org/

[72] Kernel-based Virtual Machine (Qumranet). Accessed on 19/05/2009. [Online]. Available: http://kvm.qumranet.com/kvmwiki

[73] J. Rutkowska, "Red Pill... or how to detect VMM using (almost) one CPU instruction," Tech. Rep., November 2004. [Online]. Available: http://invisiblethings.org/papers/redpill.html

[74] P. Ferrie, "Attacks on Virtual Machine Emulators," Tech. Rep., November 2004. [Online]. Available: http://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf

[75] VMware API Scripting. Accessed on 19/05/2009. [Online]. Available: http://www.vmware.com/support/developer/vix-api/

[76] Programmeringsspråk - PHP. Accessed on 19/05/2009. [Online]. Available: http://www.php.net

[77] ASP (Active Server Pages). Accessed on 19/05/2009. [Online]. Available: http://www.microsoft.com/

[78] Relasjonsdatabase - SQLite. Accessed on 19/05/2009. [Online]. Available: http://www.sqlite.org

[79] MySQL. Accessed on 19/05/2009. [Online]. Available: http://www.mysql.com/

[80] PostgreSQL. Accessed on 19/05/2009. [Online]. Available: http://www.postgresql.org/

[81] Pattern - Transfer Object. Accessed on 19/05/2009. [Online]. Available: http://java.sun.com/blueprints/corej2eepatterns/Patterns/TransferObject.html

[82] Microsoft Security Bulletin MS08-067. Accessed on 19/05/2009. [Online]. Available: http://www.microsoft.com/technet/security/bulletin/MS08-067.mspx

[83] Gonzor Switchblade. Accessed on 19/05/2009. [Online]. Available: http://gonzor228.com/

A Autoscript.sh

```
#!/bin/bash

if [ -n "$1" ]; then
    FOLDER="$1/"
    mkdir $FOLDER
fi

MY_DATE=`date +%d-%m-%Y_%H.%M.%S`
PROCMON_LOGFILE="${FOLDER}procmon_$MY_DATE.csv"
TCPDUMP_LOGFILE="${FOLDER}tcpdump_$MY_DATE.txt"
VOLATILE_LOGFILE="${FOLDER}volatile_$MY_DATE.txt"
MACTIME_LOGFILE="${FOLDER}mactime_$MY_DATE.txt"
AFICK_LOGFILE="${FOLDER}afick_$MY_DATE.txt"

# Tilbakestiller maskin til "ren" tilstand
echo -e "----> Reverterer til snapshot\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox revertToSnapshot \
"[standard] Victim_WinXP/Victim_WinXP.vmx" victim-safe

echo -e "----> Starter OS:\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox start \
"[standard] Victim_WinXP/Victim_WinXP.vmx"

# Starte logging av nettverkstrafikk
echo -e "----> Starter TCPdump\n"
tcpdump -i vmnet1 -A -tttt -s0 > $TCPDUMP_LOGFILE &


# Starte logg-agent paa maalmaskin
echo -e "----> starter agent for logging\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp "test
    " runProgramInGuest \
"[standard] Victim_WinXP/Victim_WinXP.vmx" "C:\agent\start.bat"

# Trykk tast for aa fortsette
read -p "Logging er startet. Trykk en tast naar Exploit er ferdig"

#################################
#<----   Start exploit   ---->  #
#################################

# Stop logging
echo -e "----> stopper logging...\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp "test
    " runProgramInGuest \
"[standard] Victim_WinXP/Victim_WinXP.vmx" "C:\agent\stop.bat"

# Stop TCPdump
echo -e "----> Dreper TCPdump\n"
killall tcpdump
```

```
# Overfoer logg (PROCMON-LOG) til analyse-path
echo -e "----> Overfoerer logger\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp "test
    " copyFileFromGuestToHost \
"[standard] Victim_WinXP/Victim_WinXP.vmx" "C:\agent\log.csv" /home/
    experiment/$PROCMON_LOGFILE

#################################
#   STATISK ANALYSE     #
#################################

# Suspende gjeste-OS
echo -e "----> Suspender OS\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox suspend \
"[standard] Victim_WinXP/Victim_WinXP.vmx"
sleep 5

# Hente ut MINNE-informasjon
echo -e "----> starter Volatile\n"
python /home/experiment/Volatility-1.1.2/volatility.py pslist -f /var/lib
    /vmware/Virtual\ Machines/Victim_WinXP/Victim_WinXP*.vmem > /tmp/
    pslist_old.txt

# Sammenligne prosesser foer og etter exploit
diff pslist_CLEAN.txt /tmp/pslist_old.txt > $VOLATILE_LOGFILE

# POWER-OFF (maa foerst starte fra suspend og stoppe den igjen)
echo -e "----> Resumer fra sleep\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox start \
"[standard] Victim_WinXP/Victim_WinXP.vmx"

echo -e "----> Shutdown OS:\n"
vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox stop \
"[standard] Victim_WinXP/Victim_WinXP.vmx"

# Mounte filsystem
echo -e "----> mounter FS\n"
vmware-mount /var/lib/vmware/Virtual\ Machines/Victim_WinXP/Victim_WinXP
    -*.vmdk /mnt/victim_XP/

# Hente ut MD5 og MAC-times
echo -e "----> Kjoerer mac-robber og mactime\n"
mac-robber /mnt/victim_XP/ > /tmp/mac_res.body &&  mactime -b /tmp/
    mac_res.body `date +%m/%d/%Y` > $MACTIME_LOGFILE
rm /tmp/mac_res.body

cp /var/lib/afick/afick_victim_clean /var/lib/afick/afick
afick -c /etc/afick.conf -k > $AFICK_LOGFILE

# Unmounte filsystem
echo -e "----> Unmounter FS\n"
vmware-mount -d /mnt/victim_XP
```

```
echo -e "\n\n\t DONE!"

#########################
#   ANALYSE     #
#########################
rm /svn/trunk/logs/procmon-logger/log.csv
rm /svn/trunk/logs/tcpdump-logger/log.ascii

ln -s /home/experiment/$PROCMON_LOGFILE  /svn/trunk/logs/procmon-logger/
    log.csv
ln -s /home/experiment/$TCPDUMP_LOGFILE  /svn/trunk/logs/tcpdump-logger/
    log.ascii
#ln -s /home/experiment/log/$VOLATILE_LOGFILE  /svn/trunk/logs/xxxxxxxx/
    log.txt
#ln -s /home/experiment/log/$MACTIME_LOGFILE  /svn/trunk/logs/xxxxxxxxx/
    log.txt
#ln -s /home/experiment/log/$AFICK_LOGFILE  /svn/trunk/logs/xxxxxxxxx/log
    .txt

rm /dev/shm/data.dta 2>/dev/null

#Importerer databasedefinisjoner
sqlite /dev/shm/data.dta < /svn/trunk/data.sql

cd /svn/trunk/

# Leser logger og leggr inn i databasen
./main.php
cd /home/experiment

rm /svn/trunk/logs/procmon-logger/log.csv
rm /svn/trunk/logs/tcpdump-logger/log.ascii
```

B Agent

B.1 start.bat

```
:: STARTE LOGGING

::Starter process monitor minimert og logger til fil
start C:\agent\procmon.exe /AcceptEula /NoFilter /quiet /minimized /
    backingfile C:\agent\log.pml

::Vent til procmon er klar
C:\agent\procmon.exe /AcceptEula /WaitForIdle
```

B.2 stop.bat

```
:: STOPPE LOGGING

::Avslutter process monitor og lagrer . pml log
C:\agent\procmon.exe /AcceptEula /terminate
::Aapner process monitor log og konverterer den til CSV format
C:\agent\procmon.exe /AcceptEula /NoFilter /OpenLog C:\agent\log.pml /
    SaveAs C:\agent\log.csv
```

C Fiktiv exploit

C.1 fiktiv-pre.bat

```
:: PRE-FIKTIV EXPLOIT - Filer, registernoekkel og prosess som finnes foer
     start av fiktiv exploit
echo fil1.txt > C:\fil1.txt
echo fil2.txt > C:\fil2.txt
echo fil3.txt > C:\fil3.txt
C:\Windows\system32\REG ADD HKLM\Software\TestCompany /v User /t REG_SZ /
    d best /f
start C:\Windows\system32\notepad.exe
```

C.2 fiktiv.bat

```
:: START FIKTIV EXPLOIT
::Starte prosess
start C:\Windows\system32\calc.exe

::Avslutter prosess
C:\Windows\system32\tskill notepad

::Skrive til ny, aapne, endre og slette fil
echo Fiktiv > C:\windows\system32\fiktiv.txt
type C:\fil1.txt
echo endring >> C:\fil2.txt
del C:\fil3.txt

::Laste ned fil fra internett
C:\wget.exe http://hovedprosjekter.hig.no/v2009/imt/is/kse/data/themes/
    kse/img/kse-logo.png

::Skriv til, lese fra, endre og slette register
C:\Windows\system32\REG ADD HKLM\Software\TestCompany /v Keyname /t
    REG_SZ /d 111-22 /f
C:\Windows\system32\REG QUERY "HKLM\SOFTWARE\Microsoft\Windows\
    CurrentVersion\RunOnce" /s
C:\Windows\system32\REG ADD "HKCU\Control Panel\Desktop" /v
    PaintDesktopVersion /t REG_DWORD /d 0x00000001 /f
C:\Windows\system32\REG DELETE HKLM\Software\TestCompany /v User /f
```

D monitor.cs

```
using System;
using System.IO;

namespace filemonitor {
    class Program {
        static void Main(string[] args) {
            FileSystemWatcher watcher = new FileSystemWatcher();

            // Angi hvilken katalog som skal overvakes
            watcher.Path = "c:\\temp";

            // Angi hva som skal overvakes
            watcher.Changed += new FileSystemEventHandler( OnChanged );
            watcher.Created += new FileSystemEventHandler( OnChanged );
            watcher.Deleted += new FileSystemEventHandler( OnChanged );
            watcher.Renamed += new RenamedEventHandler( OnRenamed );

            // Start overvaking
            watcher.EnableRaisingEvents = true;

            Console.ReadKey();
        }

        // Eventhandler for endringer
        private static void OnChanged(object source, FileSystemEventArgs
            e) {
            Console.WriteLine("File: " + e.FullPath + " " + e.ChangeType)
                ;
        }

        // Eventhandler for omdoping av filer
        private static void OnRenamed(object source, RenamedEventArgs e)
            {
            Console.WriteLine("File: {0} renamed to {1}", e.OldFullPath,
                e.FullPath);
        }
    }
}
```

E Kode fra Webside for presentasjon

Klassedefinisjoner

```php
<?php

/**
 * DTO-klasse for en filhendelse
 */
class FileEvent {
    public $date;           /* Dato */
    public $operation;      /* Om filen er lest, skrevet, etc. */
    public $filename;       /* Filen operasjonen er utfort pa */
    public $starterProcess; /* Hvilken program som utforte operasjonen */
    public $isFolder;       /* Er filen en katalog? */
    public $offset;         /* Hvor i filen begynte operasjonen? */
    public $length;         /* Hvor mye ble lest eller skrevet? */
    public $falsePoints;    /* Poeng som gis hvis eventen er en antatt falsk positiv */
    public $riskPoints;     /* Poeng som gis hvis eventen er en antatt risiko */
}


/**
 * DTO-klasse for en nettverkehendlse
 *
 * Et objekt representerer en nettverkspakke
 */
class IpNetworkEvent {
    public $date;
    public $srcIp;          /* IP-adresse pa avsender */
    public $dstIp;          /* Port pa avsender */
    public $srcPort;        /* IP-adresse pa mottaker */
    public $dstPort;        /* Port pa mottaker */
    public $data;           /* Pakkens data */
    public $falsePoints;
    public $riskPoints;
}


/**
 * DTO-klasse for en prosesshendelse
 */
class ProcessEvent {
    public $date;
    public $operation;
    public $filename;
    public $starterProcess;
    public $pid;            /* Prosessens ID */
    public $parentPid;      /* Foreldreprosessens ID */
    public $userTime;       /* Hvor mye usertime prosessen har brukt */
    public $kernelTime;     /* Hvor mye kerneltime prosessen har brukt */
    public $falsePoints;
    public $riskPoints;
}


/**
 * DTO-klasse for en registerhendelse
 */
class RegisterEvent {
    public $date;
    public $operation;
    public $starterProcess;
    public $key;            /* Navn pa registernokkel */
    public $value;          /* Verdi pa registernokkel */
    public $type;           /* Hvilken type registernokkelen har */
    public $falsePoints;
    public $riskPoints;
}

?>
```

Startfil

```php
<?php

/**
 * Include Data Transfer Objects
 */
include_once( "dto/file-event.class.php" );
include_once( "dto/process-event.class.php" );
include_once( "dto/ip-network-event.class.php" );
include_once( "dto/register-event.class.php" );


/**
 * Include Data Access Objects
 */
include_once( "datastore/sqlite-file-event-dao.class.php" );
include_once( "datastore/sqlite-process-event-dao.class.php" );
include_once( "datastore/sqlite-register-event-dao.class.php" );
include_once( "datastore/sqlite-ip-network-event-dao.class.php" );
include_once( "datastore/sqlite-factory.php" );


/**
 * Include Filters
 */
include_once( "filters/insert-filter.php" );


/**
 * Main entry point
 */
$sqliteFactory = new SQLiteFactory();

$fileEventDAO      = $sqliteFactory->getFileEventDAO();
$processEventDAO   = $sqliteFactory->getProcessEventDAO();
$registerEventDAO  = $sqliteFactory->getRegisterEventDAO();
$ipNetworkEventDAO = $sqliteFactory->getIpNetworkEventDAO();


include_once( "lese-fra-procmon.php" );
include_once( "lese-fra-tcpdump.php" );


?>
```

SQL-tabeller

```
/**
 * Tabeller
 */
CREATE TABLE FileEvents (
    Date            datetime,
    Operation       varchar(50),
    Filename        varchar(255),
    StarterProcess  varchar(50),
    IsFolder        int,
    Offset          varchar(20),
    Length          varchar(20),
    falsePoints     int,
    riskPoints      int
);


CREATE TABLE IpNetworkEvents (
    Date            timestamp,
    SrcIp           varchar(15),
    DstIp           varchar(15),
    SrcPort         bigint,
    DstPort         bigint,
    Data            text,
    falsePoints     int,
    riskPoints      int
);


CREATE TABLE ProcessEvents (
    Date            timestamp,
    Operation       varchar(50),
    Filename        varchar(255),
    StarterProcess  varchar(50),
    Pid             int,
    ParentPid       int,
    UserTime        double,
    KernelTime      double,
    falsePoints     int,
    riskPoints      int
);


CREATE TABLE RegisterEvents (
    Date            timestamp,
    Operation       varchar(50),
    Filename        varchar(255),
    StarterProcess  varchar(50),
    Key             varchar(255),
    Value           text,
    Type            varchar(50),
    falsePoints     int,
```

```
    riskPoints     int
);


/**
 * Views
 */
CREATE VIEW allTables AS
    SELECT date, falsePoints, riskPoints FROM FileEvents
UNION ALL
    SELECT date, falsePoints, riskPoints FROM RegisterEvents
UNION ALL
    SELECT date, falsePoints, riskPoints FROM IpNetworkEvents
UNION ALL
    SELECT date, falsePoints, riskPoints FROM ProcessEvents;


CREATE VIEW IpConnections AS
    select distinct SrcIp, SrcPort, DstIp, DstPort from IpNetworkEvents;
```

F Rapport for fiktiv exploit

Sammendrag analyse

Navn:	Test av exploit nr 4
Test start	2009-04-30 11:45:07
Test slutt	2009-05-07 11:45:24

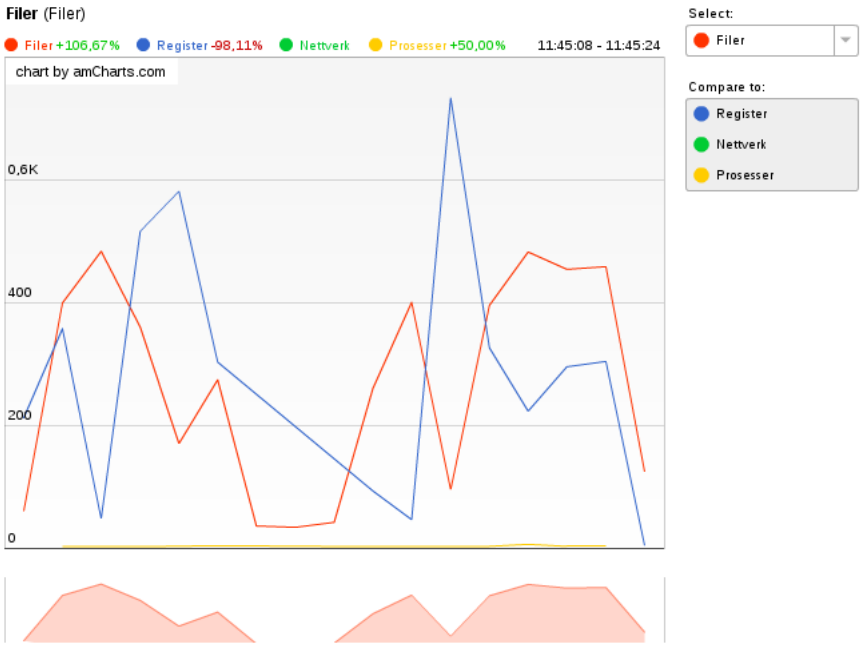
Oppsummering endinger

	Lest	Opprettet	Skrevet	Slettet
Filer	799	280	4	1
Register	1300	106	47	1

	Opprettet	Startet	Avsluttet
Prosesser	10	9	8

	Opprettet
Nettverks-tilkoblinger	3

Tidslinje



F.1 Afick

```
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/CALC.EXE-02CD573A.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/TSKILL.EXE-2F6AAB7F.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WGET.EXE-2ACD219E.pf
new unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/Logs/tmp.edb
new unknown_type : /mnt/victim_XP/WINDOWS/Temp/Perflib_Perfdata_7d4.dat
new unknown_type : /mnt/victim_XP/WINDOWS/system32/fiktiv.txt
new unknown_type : /mnt/victim_XP/agent/log.csv
new unknown_type : /mnt/victim_XP/agent/log.pml
new unknown_type : /mnt/victim_XP/fiktiv.bat
deleted unknown_type : /mnt/victim_XP/fil3.txt
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/Application Data/IconCache.
     db
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/NTUSER.DAT
changed unknown_type : /mnt/victim_XP/System Volume Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB
     }/_driver.cfg
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/CMD.EXE-087B4001.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/PROCMON.EXE-09DDAFE0.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/REG.EXE-0D2A95F7.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/VMIP.EXE-2BD5723A.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WMIPRVSE.EXE-28F301A9.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WSCNTFY.EXE-1B24F5EB.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WUAUCLT.EXE-399A8E72.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/SchedLgU.Txt
changed unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/DataStore.edb
changed unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/Logs/edb.chk
changed unknown_type : /mnt/victim_XP/WINDOWS/Tasks/SA.DAT
changed unknown_type : /mnt/victim_XP/WINDOWS/bootstat.dat
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/AppEvent.Evt
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/SysEvent.Evt
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/software
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/system
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/INDEX.BTR
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/MAPPING1.MAP
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/OBJECTS.DATA
changed unknown_type : /mnt/victim_XP/fil2.txt
changed unknown_type : /mnt/victim_XP/pagefile.sys

# Hash database : 9192 files scanned, 34 changed (new : 9; delete : 1; changed : 24; dangling : 0;
     exclude_suffix : 1792; exclude_prefix : 0; exclude_re : 0; degraded : 11)
# ############################################################
# MD5 hash of /var/lib/afick/afick => srk3IkrmAC3PSrKIz3HlWw
# user time : 16.52; system time : 11.12; real time : 167
```

F.2 MAC-times

```
...
Thu May 07 2009 14:40:46      252 .a. -r-------- 0        0        10932    /mnt/victim_XP/agent/start.bat
                                0 .a. dr-x------ 0        0        10498    /mnt/victim_XP/Program Files/
                                      VMware
                             8192 m.c -r-------- 0        0        3514     /mnt/victim_XP/WINDOWS/system32/
                                      config/SAM.LOG
                           640000 .a. -r-------- 0        0        2289     /mnt/victim_XP/WINDOWS/system32/
                                      dbghelp.dll
                           290816 .a. -r-------- 0        0        2094     /mnt/victim_XP/WINDOWS/system32/
                                      winsrv.dll
                           276992 .a. -r-------- 0        0        2066     /mnt/victim_XP/WINDOWS/system32/
                                      comdlg32.dll
Thu May 07 2009 14:40:47    16896 .a. -r-------- 0        0        5610     /mnt/victim_XP/WINDOWS/system32/
     fltlib.dll
                            18944 .a. -r-------- 0        0        4750     /mnt/victim_XP/WINDOWS/system32/
                                      wbem/wbemprox.dll
                             4096 .a. dr-x------ 0        0        32       /mnt/victim_XP/WINDOWS/system
                            36864 .a. dr-x------ 0        0        61       /mnt/victim_XP/WINDOWS/system32/
                                      wbem
                         19351764 .a. -r-------- 0        0        10999    /mnt/victim_XP/agent/log.pml
                                0 .a. dr-x------ 0        0        10607    /mnt/victim_XP/Program Files/
                                      VMware/VMware Tools/Drivers/memctl
                            28672 mac dr-x------ 0        0        31       /mnt/victim_XP/WINDOWS/system32/
                                      drivers
                          2902376 ..c -r-------- 0        0        10930    /mnt/victim_XP/agent/Procmon.exe
                           214528 .a. -r-------- 0        0        4745     /mnt/victim_XP/WINDOWS/system32/
                                      wbem/wbemcomn.dll
                                0 .a. dr-x------ 0        0        10500    /mnt/victim_XP/Program Files/
                                      VMware/VMware Tools/Drivers
Thu May 07 2009 14:40:48    43520 .a. -r-------- 0        0        4751     /mnt/victim_XP/WINDOWS/system32/
     wbem/wbemsvc.dll
                           472064 .a. -r-------- 0        0        4725     /mnt/victim_XP/WINDOWS/system32/
                                      wbem/fastprox.dll
```

```
                              0 .a. dr-x------ 0        0        62       /mnt/victim_XP/WINDOWS/system32/
                                   wbem/Repository
                         378368 .a. -r-------- 0        0        3137     /mnt/victim_XP/WINDOWS/system32/
                              wzcdlg.dll
                           4096 .a. dr-x------ 0        0        30       /mnt/victim_XP/WINDOWS/system32/
                                config
                          67072 .a. -r-------- 0        0        2131     /mnt/victim_XP/WINDOWS/system32/
                               ntdsapi.dll
                         413696 .a. -r-------- 0        0        2189     /mnt/victim_XP/WINDOWS/system32/
                              msvcp60.dll
                              0 .a. dr-x------ 0        0        43       /mnt/victim_XP/WINDOWS/system32/
                                   drivers/etc
                            734 .a. -r-------- 0        0        236      /mnt/victim_XP/WINDOWS/system32/
                                    drivers/etc/hosts
                         218112 .a. -r-------- 0        0        4767     /mnt/victim_XP/WINDOWS/system32/
                              wbem/wmiprvse.exe
                           4096 .a. dr-x------ 0        0        5395     /mnt/victim_XP/WINDOWS/system32/
                                wbem/Repository/FS
Thu May 07 2009 14:40:49       4096 .a. dr-x------ 0        0        42       /mnt/victim_XP/WINDOWS/repair
                           4096 .a. dr-x------ 0        0        9999     /mnt/victim_XP/WINDOWS/system32/
                                config/systemprofile/Start Menu/Programs/Accessories
                              0 .a. dr-x------ 0        0        9996     /mnt/victim_XP/WINDOWS/system32/
                                   config/systemprofile/Start Menu
                           6144 .a. -r-------- 0        0        2091     /mnt/victim_XP/WINDOWS/system32/
                                csrss.exe
                           4096 .a. dr-x------ 0        0        144      /mnt/victim_XP/WINDOWS/system32/
                                wbem/Logs
                           4096 .a. dr-x------ 0        0        3518     /mnt/victim_XP/Documents and
                                Settings
                           4096 .a. dr-x------ 0        0        3698     /mnt/victim_XP/Documents and
                                Settings/Default User/Start Menu/Programs
                          49152 .a. dr-x------ 0        0        3519     /mnt/victim_XP/Documents and
                                Settings/Default User
                              0 .a. dr-x------ 0        0        3695     /mnt/victim_XP/Documents and
                                   Settings/Default User/Start Menu
                          50688 .a. -r-------- 0        0        2061     /mnt/victim_XP/WINDOWS/system32/
                               smss.exe
                           4096 .a. dr-x------ 0        0        9997     /mnt/victim_XP/WINDOWS/system32/
                                config/systemprofile/Start Menu/Programs
                           4096 .a. dr-x------ 0        0        9994     /mnt/victim_XP/WINDOWS/system32/
                                config/systemprofile
                           4096 .a. dr-x------ 0        0        6386     /mnt/victim_XP/Documents and
                                Settings/Default User/Start Menu/Programs/Accessories
Thu May 07 2009 14:40:50     108032 .a. -r-------- 0        0        2121     /mnt/victim_XP/WINDOWS/system32/
     services.exe
                          92672 .a. -r-------- 0        0        2480     /mnt/victim_XP/WINDOWS/system32/
                               dskquota.dll
                          13312 .a. -r-------- 0        0        2122     /mnt/victim_XP/WINDOWS/system32/
                               lsass.exe
                         144896 .a. -r-------- 0        0        4765     /mnt/victim_XP/WINDOWS/system32/
                              wbem/wmiprov.dll
                           5632 .a. -r-------- 0        0        2273     /mnt/victim_XP/WINDOWS/system32/
                                wmi.dll
                          36352 .a. -r-------- 0        0        2127     /mnt/victim_XP/WINDOWS/system32/
                               ncobjapi.dll
                           5632 .a. -r-------- 0        0        2959     /mnt/victim_XP/WINDOWS/system32/
                                security.dll
                         172032 .a. -r-------- 0        0        2079     /mnt/victim_XP/WINDOWS/system32/
                              wldap32.dll
                          14336 .a. -r-------- 0        0        2145     /mnt/victim_XP/WINDOWS/system32/
                               svchost.exe
                        1352192 .a. -r-------- 0        0        4722     /mnt/victim_XP/WINDOWS/system32/
                             wbem/cimwin32.dll
                         346672 .a. -r-------- 0        0        10712    /mnt/victim_XP/Program Files/
                              VMware/VMware Tools/vmacthlp.exe
                          16896 .a. -r-------- 0        0        2167     /mnt/victim_XP/WINDOWS/system32/
                               cfgmgr32.dll
                          31232 .a. -r-------- 0        0        1500     /mnt/victim_XP/WINDOWS/system32/
                                traffic.dll
                         185856 .a. -r-------- 0        0        4726     /mnt/victim_XP/WINDOWS/system32/
                              wbem/framedyn.dll
                         144896 .a. -r-------- 0        0        2132     /mnt/victim_XP/WINDOWS/system32/
                              schannel.dll
                          95232 .a. -r-------- 0        0        4770     /mnt/victim_XP/WINDOWS/system32/
                               wbem/wmiutils.dll
Thu May 07 2009 14:40:51       0 .a. dr-x------ 0        0        3600     /mnt/victim_XP/Documents and
     Settings/All Users/Application Data
Thu May 07 2009 14:40:52       4096 .a. dr-x------ 0        0        140      /mnt/victim_XP/WINDOWS/pchealth/
     helpctr
                            602 mac -r-------- 0        0        5393     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Logs/wmiprov.log
                           4096 .a. dr-x------ 0        0        141      /mnt/victim_XP/WINDOWS/pchealth/
                                helpctr/binaries
```

```
                            3196 mac -r-------- 0        0        5397     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/MAPPING1.MAP
                               0 .a. dr-x------ 0        0        139      /mnt/victim_XP/WINDOWS/pchealth
Thu May 07 2009 14:40:53    16896 .a. -r-------- 0        0        2152     /mnt/victim_XP/WINDOWS/system32/
     winrnr.dll
                            7116 .a. -r-------- 0        0        1397     /mnt/victim_XP/WINDOWS/system32/
                                 drivers/etc/services
                            4096 .a. dr-x------ 0        0        10163    /mnt/victim_XP/System Volume
                                 Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}
                          245248 .a. -r-------- 0        0        2147     /mnt/victim_XP/WINDOWS/system32/
                                 mswsock.dll
                          344064 .a. -r-------- 0        0        2278     /mnt/victim_XP/WINDOWS/system32/
                                 hnetcfg.dll
                            4096 .a. dr-x------ 0        0        10483    /mnt/victim_XP/System Volume
                                 Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}/RP2/snapshot/
                                 Repository/FS
                           57856 .a. -r-------- 0        0        2162     /mnt/victim_XP/WINDOWS/system32/
                                 spoolsv.exe
                           19968 .a. -r-------- 0        0        2148     /mnt/victim_XP/WINDOWS/system32/
                                 wshtcpip.dll
                               0 .a. dr-x------ 0        0        10481    /mnt/victim_XP/System Volume
                                     Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}/RP2/snapshot/
                                     Repository
                            4096 .a. dr-x------ 0        0        10464    /mnt/victim_XP/System Volume
                                 Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}/RP2/snapshot
                            8192 .a. -r-------- 0        0        2153     /mnt/victim_XP/WINDOWS/system32/
                                 rasadhlp.dll
                           57666 mac -r-------- 0        0        10239    /mnt/victim_XP/WINDOWS/Prefetch/
                                 WUAUCLT.EXE-399A8E72.pf
                            8192 .a. dr-x------ 0        0        10462    /mnt/victim_XP/System Volume
                                 Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}/RP2
Thu May 07 2009 14:40:54   539184 .a. -r-------- 0        0        10558    /mnt/victim_XP/Program Files/
     VMware/VMware Tools/VMwareService.exe
                               0 .a. dr-x------ 0        0        39       /mnt/victim_XP/WINDOWS/system32/
                                     spool/prtprocs/w32x86
                               0 .a. dr-x------ 0        0        38       /mnt/victim_XP/WINDOWS/system32/
                                     spool/prtprocs
                               0 .a. dr-x------ 0        0        10583    /mnt/victim_XP/WINDOWS/WinSxS/
                                     x86_Microsoft.VC80.MFC_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_3bf8fa05
                               0 .a. dr-x------ 0        0        34       /mnt/victim_XP/WINDOWS/system32/
                                     spool
                          830000 .a. -r-------- 0        0        10560    /mnt/victim_XP/Program Files/
                                 VMware/VMware Tools/VMwareUser.exe
                         1032192 .a. -r-------- 0        0        2218     /mnt/victim_XP/WINDOWS/explorer.
                                 exe
                            4096 .a. dr-x------ 0        0        10593    /mnt/victim_XP/WINDOWS/WinSxS/
                                 x86_Microsoft.VC80.MFCLOC_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_91481303
                          416304 .a. -r-------- 0        0        10559    /mnt/victim_XP/Program Files/
                                 VMware/VMware Tools/VMwareTray.exe
                           44544 .a. -r-------- 0        0        2335     /mnt/victim_XP/WINDOWS/system32/
                                 alg.exe
Thu May 07 2009 14:40:55    13824 .a. -r-------- 0        0        3121     /mnt/victim_XP/WINDOWS/system32/
     wscntfy.exe
                           32256 .a. -r-------- 0        0        3118     /mnt/victim_XP/WINDOWS/system32/
                                 wpabaln.exe
                           69120 .a. -r-------- 0        0        2288     /mnt/victim_XP/WINDOWS/system32/
                                 notepad.exe
Thu May 07 2009 14:40:56     1024 .a. -r-------- 0        0        3501     /mnt/victim_XP/WINDOWS/system32/
     config/system.LOG
Thu May 07 2009 14:40:57     1024 m.c -r-------- 0        0        3501     /mnt/victim_XP/WINDOWS/system32/
     config/system.LOG
Thu May 07 2009 14:40:59  5332992 mac -r-------- 0        0        5402     /mnt/victim_XP/WINDOWS/system32/
     wbem/Repository/FS/OBJECTS.DATA
                            4096 .a. dr-x------ 0        0        10256    /mnt/victim_XP/Documents and
                                 Settings/test/Local Settings
Thu May 07 2009 14:41:00      568 mac -r-------- 0        0        5400     /mnt/victim_XP/WINDOWS/system32/
     wbem/Repository/FS/INDEX.MAP
                            2628 mac -r-------- 0        0        5401     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/OBJECTS.MAP
                               4 mac -r-------- 0        0        5399     /mnt/victim_XP/WINDOWS/system32/
                                     wbem/Repository/FS/MAPPING.VER
                            3196 mac -r-------- 0        0        5398     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/MAPPING2.MAP
                           31228 mac -r-------- 0        0        10226    /mnt/victim_XP/WINDOWS/Prefetch/
                                 WMIPRVSE.EXE-28F301A9.pf
                         1114112 mac -r-------- 0        0        5403     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/INDEX.BTR
Thu May 07 2009 14:41:03      833 .a. -r-------- 0        0        11000    /mnt/victim_XP/fiktiv.bat
                          264582 mac -r-------- 0        0        10971    /mnt/victim_XP/System Volume
                                 Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}/RP4/change.log
Thu May 07 2009 14:41:10   114688 .a. -r-------- 0        0        4983     /mnt/victim_XP/WINDOWS/system32/
     calc.exe
                           16384 .a. -r-------- 0        0        4968     /mnt/victim_XP/WINDOWS/system32/
```

```
                                tskill.exe
                              477 .a. -r-------- 0        0        226      /mnt/victim_XP/WINDOWS/win.ini
Thu May 07 2009 14:41:11       21 mac -r-------- 0        0        7218     /mnt/victim_XP/fil2.txt
                            50176 .a. -r-------- 0        0        2918     /mnt/victim_XP/WINDOWS/system32/
                                reg.exe
                                9 mac -r-------- 0        0        11001    /mnt/victim_XP/WINDOWS/system32/
                                    fiktiv.txt
                             7018 mac -r-------- 0        0        11002    /mnt/victim_XP/WINDOWS/Prefetch/
                                 WGET.EXE-2ACD219E.pf
                            10742 mac -r-------- 0        0        7400     /mnt/victim_XP/WINDOWS/Prefetch/
                                TSKILL.EXE-2F6AAB7F.pf
                           311296 mac dr-x------ 0        0        29       /mnt/victim_XP/WINDOWS/system32
                           162816 .a. -r-------- 0        0        10920    /mnt/victim_XP/wget.exe
                               11 .a. -r-------- 0        0        6648     /mnt/victim_XP/fil1.txt
                            22528 .a. -r-------- 0        0        2192     /mnt/victim_XP/WINDOWS/system32/
                                wsock32.dll
Thu May 07 2009 14:41:13     8192 mac -r-------- 0        0        10237    /mnt/victim_XP/WINDOWS/
     SoftwareDistribution/DataStore/Logs/edb.chk
                             9916 mac -r-------- 0        0        10972    /mnt/victim_XP/WINDOWS/Prefetch/
                                 REG.EXE-0D2A95F7.pf
                           114688 ..c -r-------- 0        0        4983     /mnt/victim_XP/WINDOWS/system32/
                                calc.exe
Thu May 07 2009 14:41:18      276 .a. -r-------- 0        0        10934    /mnt/victim_XP/agent/stop.bat
...
```

G Rapport for kjent exploit

Sammendrag analyse

Navn:	Test av exploit nr 4
Test start	2009-05-08 15:28:29
Test slutt	2009-05-08 15:29:10

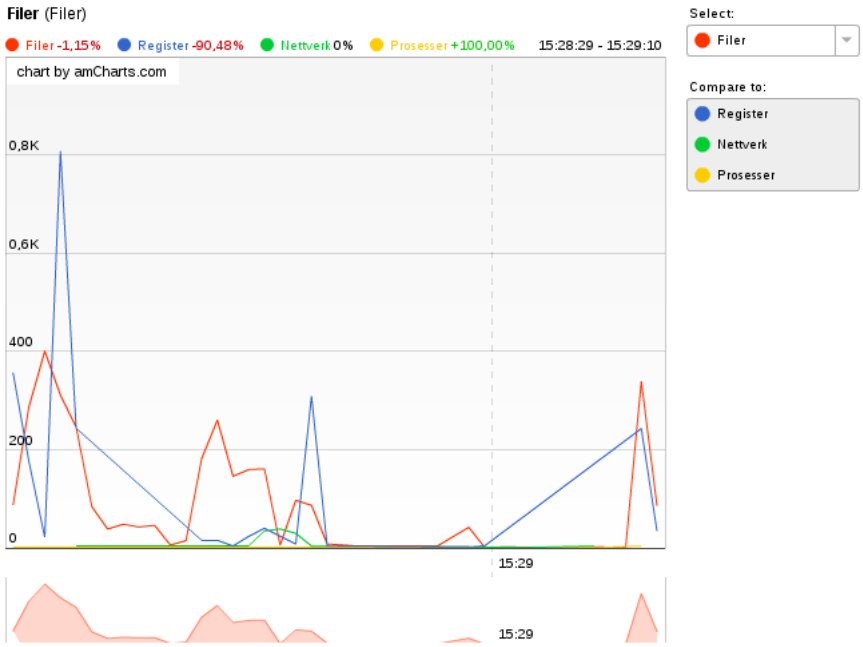
Oppsummering endinger

	Lest	Opprettet	Skrevet	Slettet
Filer	398	98	52	
Register	748	38	19	

	Opprettet	Startet	Avsluttet
Prosesser	3	3	3

	Opprettet
Nettverks-tilkoblinger	13

Tidslinje



G.1 Afick

```
new unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/Logs/tmp.edb
new unknown_type : /mnt/victim_XP/WINDOWS/Temp/Perflib_Perfdata_7d4.dat
new unknown_type : /mnt/victim_XP/agent/log.csv
new unknown_type : /mnt/victim_XP/agent/log.pml
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/Application Data/IconCache.
     db
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/NTUSER.DAT
changed unknown_type : /mnt/victim_XP/System Volume Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB
     }/_driver.cfg
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/CMD.EXE-087B4001.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/PROCMON.EXE-09DDAFE0.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/VMIP.EXE-2BD5723A.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WMIPRVSE.EXE-28F301A9.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WSCNTFY.EXE-1B24F5EB.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/SchedLgU.Txt
changed unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/DataStore.edb
changed unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/Logs/edb.chk
changed unknown_type : /mnt/victim_XP/WINDOWS/Tasks/SA.DAT
changed unknown_type : /mnt/victim_XP/WINDOWS/bootstat.dat
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/CatRoot2/edb.chk
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/CatRoot2/{F750E6C3-38EE-11D1-85E5-00C04FC295EE}/catdb
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/AppEvent.Evt
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/SAM
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/SysEvent.Evt
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/software
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/system
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/INDEX.BTR
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/INDEX.MAP
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/MAPPING1.MAP
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/MAPPING2.MAP
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/OBJECTS.DATA
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/OBJECTS.MAP
changed unknown_type : /mnt/victim_XP/pagefile.sys

# Hash database : 9192 files scanned, 31 changed (new : 4; delete : 0; changed : 27; dangling : 0;
     exclude_suffix : 1792; exclude_prefix : 0; exclude_re : 0; degraded : 11)
# ############################################################
# MD5 hash of /var/lib/afick/afick => yhx4Jl+WnKdvt5Qxb/FhQw
# user time : 16.43; system time : 11.45; real time : 265
```

G.2 MAC-times

```
...
Fri May 08 2009 15:28:27      252 .a. -r-------- 0        0        10932    /mnt/victim_XP/agent/start.bat
                           290816 .a. -r-------- 0        0        2094     /mnt/victim_XP/WINDOWS/system32/
                                winsrv.dll
                             8192 m.c -r-------- 0        0        3514     /mnt/victim_XP/WINDOWS/system32/
                                 config/SAM.LOG
Fri May 08 2009 15:28:28  2902376 .a. -r-------- 0        0        10930    /mnt/victim_XP/agent/Procmon.exe
                           276992 .a. -r-------- 0        0        2066     /mnt/victim_XP/WINDOWS/system32/
                                comdlg32.dll
                           640000 .a. -r-------- 0        0        2289     /mnt/victim_XP/WINDOWS/system32/
                                dbghelp.dll
                            23040 .a. -r-------- 0        0        2271     /mnt/victim_XP/WINDOWS/system32/
                                 psapi.dll
Fri May 08 2009 15:28:29    43520 .a. -r-------- 0        0        4751     /mnt/victim_XP/WINDOWS/system32/
     wbem/wbemsvc.dll
                            28672 mac dr-x------ 0        0        31       /mnt/victim_XP/WINDOWS/system32/
                                drivers
                            67072 .a. -r-------- 0        0        2131     /mnt/victim_XP/WINDOWS/system32/
                                 ntdsapi.dll
                             4096 .a. dr-x------ 0        0        5395     /mnt/victim_XP/WINDOWS/system32/
                                  wbem/Repository/FS
                          2902376 ..c -r-------- 0        0        10930    /mnt/victim_XP/agent/Procmon.exe
                            16896 .a. -r-------- 0        0        5610     /mnt/victim_XP/WINDOWS/system32/
                                 fltlib.dll
                             4096 .a. dr-x------ 0        0        32       /mnt/victim_XP/WINDOWS/system
                           472064 .a. -r-------- 0        0        4725     /mnt/victim_XP/WINDOWS/system32/
                                wbem/fastprox.dll
                            36864 .a. dr-x------ 0        0        61       /mnt/victim_XP/WINDOWS/system32/
                                wbem
                         11181969 .a. -r-------- 0        0        11003    /mnt/victim_XP/agent/log.pml
                                0 .a. dr-x------ 0        0        10607    /mnt/victim_XP/Program Files/
                                    VMware/VMware Tools/Drivers/memctl
                           264926 mac -r-------- 0        0        10971    /mnt/victim_XP/System Volume
                                Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}/RP4/change.log
                           502272 .a. -r-------- 0        0        2098     /mnt/victim_XP/WINDOWS/system32/
                                winlogon.exe
```

```
                                0 .a. dr-x------ 0        0        10500    /mnt/victim_XP/Program Files/
                                    VMware/VMware Tools/Drivers
                                0 .a. dr-x------ 0        0        62       /mnt/victim_XP/WINDOWS/system32/
                                    wbem/Repository
                            18944 .a. -r-------- 0        0        4750     /mnt/victim_XP/WINDOWS/system32/
                                  wbem/wbemprox.dll
                           214528 .a. -r-------- 0        0        4745     /mnt/victim_XP/WINDOWS/system32/
                                wbem/wbemcomn.dll
Fri May 08 2009 15:28:30     4096 .a. dr-x------ 0        0        30       /mnt/victim_XP/WINDOWS/system32/
     config
                             4096 .a. dr-x------ 0        0        9994     /mnt/victim_XP/WINDOWS/system32/
                                  config/systemprofile
                           218112 .a. -r-------- 0        0        4767     /mnt/victim_XP/WINDOWS/system32/
                                wbem/wmiprvse.exe
                                0 .a. dr-x------ 0        0        3695     /mnt/victim_XP/Documents and
                                    Settings/Default User/Start Menu
                                0 .a. dr-x------ 0        0        9996     /mnt/victim_XP/WINDOWS/system32/
                                    config/systemprofile/Start Menu
                             4096 .a. dr-x------ 0        0        5351     /mnt/victim_XP/WINDOWS/
                                  Registration
                             4096 .a. dr-x------ 0        0        3698     /mnt/victim_XP/Documents and
                                  Settings/Default User/Start Menu/Programs
                             4096 .a. dr-x------ 0        0        42       /mnt/victim_XP/WINDOWS/repair
                             4096 .a. dr-x------ 0        0        9999     /mnt/victim_XP/WINDOWS/system32/
                                  config/systemprofile/Start Menu/Programs/Accessories
                            49152 .a. dr-x------ 0        0        3519     /mnt/victim_XP/Documents and
                                  Settings/Default User
                             4096 .a. dr-x------ 0        0        6386     /mnt/victim_XP/Documents and
                                  Settings/Default User/Start Menu/Programs/Accessories
                             4096 .a. dr-x------ 0        0        9997     /mnt/victim_XP/WINDOWS/system32/
                                  config/systemprofile/Start Menu/Programs
Fri May 08 2009 15:28:31    50688 .a. -r-------- 0        0        2061     /mnt/victim_XP/WINDOWS/system32/
     smss.exe
                           185856 .a. -r-------- 0        0        4726     /mnt/victim_XP/WINDOWS/system32/
                                wbem/framedyn.dll
                           172032 .a. -r-------- 0        0        2079     /mnt/victim_XP/WINDOWS/system32/
                                wldap32.dll
                            18432 .a. -r-------- 0        0        2157     /mnt/victim_XP/WINDOWS/system32/
                                  wtsapi32.dll
                            57344 .a. -r-------- 0        0        2088     /mnt/victim_XP/WINDOWS/system32/
                                  msasn1.dll
                          1352192 .a. -r-------- 0        0        4722     /mnt/victim_XP/WINDOWS/system32/
                               wbem/cimwin32.dll
                             6144 .a. -r-------- 0        0        2091     /mnt/victim_XP/WINDOWS/system32/
                                  csrss.exe
                            95232 .a. -r-------- 0        0        4770     /mnt/victim_XP/WINDOWS/system32/
                                  wbem/wmiutils.dll
                             5632 .a. -r-------- 0        0        2959     /mnt/victim_XP/WINDOWS/system32/
                                  security.dll
                           332288 .a. -r-------- 0        0        2103     /mnt/victim_XP/WINDOWS/system32/
                                netapi32.dll
                           144896 .a. -r-------- 0        0        4765     /mnt/victim_XP/WINDOWS/system32/
                                wbem/wmiprov.dll
                           597504 .a. -r-------- 0        0        2086     /mnt/victim_XP/WINDOWS/system32/
                                crypt32.dll
                            36352 .a. -r-------- 0        0        2127     /mnt/victim_XP/WINDOWS/system32/
                                  ncobjapi.dll
                           144896 .a. -r-------- 0        0        2132     /mnt/victim_XP/WINDOWS/system32/
                                schannel.dll
                            53760 .a. -r-------- 0        0        2101     /mnt/victim_XP/WINDOWS/system32/
                                  winsta.dll
                            16896 .a. -r-------- 0        0        2167     /mnt/victim_XP/WINDOWS/system32/
                                  cfgmgr32.dll
                            31232 .a. -r-------- 0        0        1500     /mnt/victim_XP/WINDOWS/system32/
                                  traffic.dll
                            92672 .a. -r-------- 0        0        2480     /mnt/victim_XP/WINDOWS/system32/
                                  dskquota.dll
                             4096 .a. dr-x------ 0        0        144      /mnt/victim_XP/WINDOWS/system32/
                                  wbem/Logs
Fri May 08 2009 15:28:32    13312 .a. -r-------- 0        0        2122     /mnt/victim_XP/WINDOWS/system32/
     lsass.exe
                             3196 mac -r-------- 0        0        5398     /mnt/victim_XP/WINDOWS/system32/
                                  wbem/Repository/FS/MAPPING2.MAP
                           108032 .a. -r-------- 0        0        2121     /mnt/victim_XP/WINDOWS/system32/
                                services.exe
Fri May 08 2009 15:28:33   344064 .a. -r-------- 0        0        2278     /mnt/victim_XP/WINDOWS/system32/
     hnetcfg.dll
                              602 mac -r-------- 0        0        5393     /mnt/victim_XP/WINDOWS/system32/
                                   wbem/Logs/wmiprov.log
                           346672 .a. -r-------- 0        0        10712    /mnt/victim_XP/Program Files/
                                VMware/VMware Tools/vmacthlp.exe
                                0 .a. dr-x------ 0        0        43       /mnt/victim_XP/WINDOWS/system32/
                                    drivers/etc
```

```
                          245248 .a. -r-------- 0        0        2147     /mnt/victim_XP/WINDOWS/system32/
                               mswsock.dll
                            7116 .a. -r-------- 0        0        1397     /mnt/victim_XP/WINDOWS/system32/
                                 drivers/etc/services
                            8192 .a. -r-------- 0        0        2153     /mnt/victim_XP/WINDOWS/system32/
                                 rasadhlp.dll
                           16896 .a. -r-------- 0        0        2152     /mnt/victim_XP/WINDOWS/system32/
                                 winrnr.dll
                           19968 .a. -r-------- 0        0        2148     /mnt/victim_XP/WINDOWS/system32/
                                 wshtcpip.dll
                           14336 .a. -r-------- 0        0        2145     /mnt/victim_XP/WINDOWS/system32/
                                 svchost.exe
Fri May 08 2009 15:28:35        0 .a. dr-x------ 0        0        139      /mnt/victim_XP/WINDOWS/pchealth
                            4096 .a. dr-x------ 0        0        140      /mnt/victim_XP/WINDOWS/pchealth/
                                 helpctr
                            4096 .a. dr-x------ 0        0        141      /mnt/victim_XP/WINDOWS/pchealth/
                                 helpctr/binaries
Fri May 08 2009 15:28:41  5332992 mac -r-------- 0        0        5402     /mnt/victim_XP/WINDOWS/system32/
     wbem/Repository/FS/OBJECTS.DATA
Fri May 08 2009 15:28:42  1114112 mac -r-------- 0        0        5403     /mnt/victim_XP/WINDOWS/system32/
     wbem/Repository/FS/INDEX.BTR
                           57856 .a. -r-------- 0        0        2162     /mnt/victim_XP/WINDOWS/system32/
                                 spoolsv.exe
                             568 mac -r-------- 0        0        5400     /mnt/victim_XP/WINDOWS/system32/
                                  wbem/Repository/FS/INDEX.MAP
                            2628 mac -r-------- 0        0        5401     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/OBJECTS.MAP
                            3196 mac -r-------- 0        0        5397     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/MAPPING1.MAP
                               4 mac -r-------- 0        0        5399     /mnt/victim_XP/WINDOWS/system32/
                                    wbem/Repository/FS/MAPPING.VER
                           31276 mac -r-------- 0        0        10226    /mnt/victim_XP/WINDOWS/Prefetch/
                                 WMIPRVSE.EXE-28F301A9.pf
Fri May 08 2009 15:28:43        0 .a. dr-x------ 0        0        39       /mnt/victim_XP/WINDOWS/system32/
     spool/prtprocs/w32x86
                         1032192 .a. -r-------- 0        0        2218     /mnt/victim_XP/WINDOWS/explorer.
                                 exe
                               0 .a. dr-x------ 0        0        34       /mnt/victim_XP/WINDOWS/system32/
                                    spool
                               0 .a. dr-x------ 0        0        38       /mnt/victim_XP/WINDOWS/system32/
                                    spool/prtprocs
Fri May 08 2009 15:28:44        0 .a. dr-x------ 0        0        10583    /mnt/victim_XP/WINDOWS/WinSxS/
     x86_Microsoft.VC80.MFC_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_3bf8fa05
                           32256 .a. -r-------- 0        0        3118     /mnt/victim_XP/WINDOWS/system32/
                                 wpabaln.exe
                           44544 .a. -r-------- 0        0        2335     /mnt/victim_XP/WINDOWS/system32/
                                 alg.exe
                          830000 .a. -r-------- 0        0        10560    /mnt/victim_XP/Program Files/
                                 VMware/VMware Tools/VMwareUser.exe
                            4096 .a. dr-x------ 0        0        10593    /mnt/victim_XP/WINDOWS/WinSxS/
                                  x86_Microsoft.VC80.MFCLOC_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_91481303
                           13824 .a. -r-------- 0        0        3121     /mnt/victim_XP/WINDOWS/system32/
                                 wscntfy.exe
                          539184 .a. -r-------- 0        0        10558    /mnt/victim_XP/Program Files/
                                 VMware/VMware Tools/VMwareService.exe
Fri May 08 2009 15:28:45   111104 .a. -r-------- 0        0        5693     /mnt/victim_XP/WINDOWS/system32/
     wuauclt.exe
                           69120 .a. -r-------- 0        0        2288     /mnt/victim_XP/WINDOWS/system32/
                                 notepad.exe
Fri May 08 2009 15:29:09      276 .a. -r-------- 0        0        10934    /mnt/victim_XP/agent/stop.bat
...
```

H Rapport for kompleks exploit

Sammendrag analyse

Navn:	Test av kompleks exploit
Test start	2009-05-15 11:56:52
Test slutt	2009-05-16 12:00:15

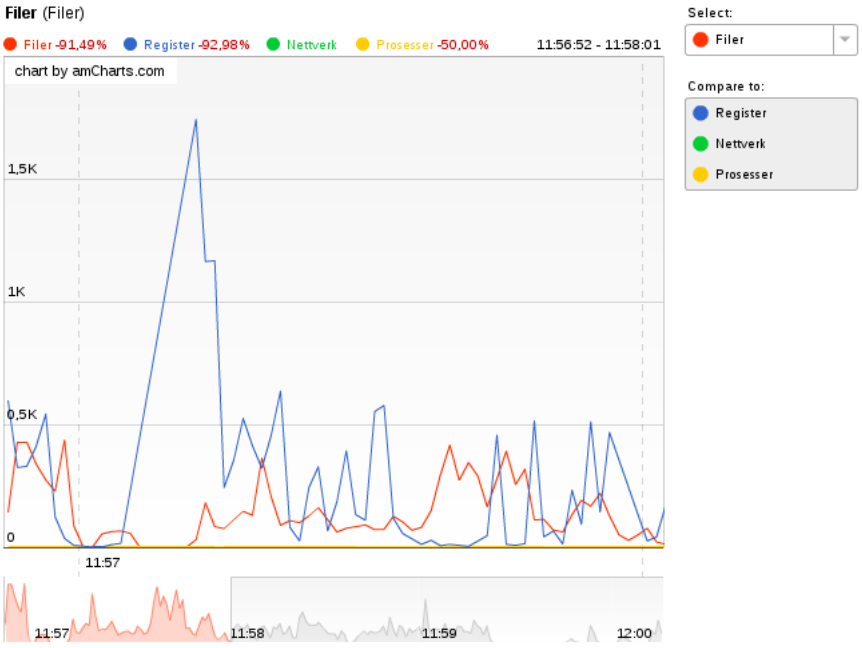
Oppsummering endinger

	Lest	Opprettet	Skrevet	Slettet
Filer	2401	3477	2790	17
Register	10404	1302	607	15

	Opprettet	Startet	Avsluttet
Prosesser	40	40	36

	Opprettet
Nettverks-tilkoblinger	7

Tidslinje



H.1 Startede prosesser - Dynamisk analyse

```
Dato                      Startprogram               Filnavn

2009-05-16 11:57:21 Explorer.EXE    C:\WINDOWS\system32\wscript.exe
2009-05-16 11:57:29 wscript.exe E:\LaunchU3.exe
2009-05-16 11:57:32 wscript.exe C:\WINDOWS\system32\cmd.exe
2009-05-16 11:57:39 cmd.exe     C:\WINDOWS\system32\ipconfig.exe
2009-05-16 11:57:46 cmd.exe     E:\SYSTEM\SRC\WGET.EXE
2009-05-16 11:57:50 LaunchU3.exe    C:\Documents and Settings\test\Application Data\U3\0000060327026187\
     LaunchPad.exe
2009-05-16 11:57:51 cmd.exe     C:\WINDOWS\system32\xcopy.exe
2009-05-16 11:57:54 cmd.exe     C:\WINDOWS\system32\sc.exe
2009-05-16 11:57:56 cmd.exe     C:\WINDOWS\system32\sc.exe
2009-05-16 11:58:04 cmd.exe     C:\WINDOWS\regedit.exe
2009-05-16 11:58:07 cmd.exe     C:\WINDOWS\system32\net.exe
2009-05-16 11:58:11 net.exe     C:\WINDOWS\system32\net1.exe
2009-05-16 11:58:12 services.exe    C:\WINDOWS\winvnc.exe
2009-05-16 11:58:16 cmd.exe     C:\WINDOWS\system32\xcopy.exe
2009-05-16 11:58:26 cmd.exe     C:\WINDOWS\system32\reg.exe
2009-05-16 11:58:28 cmd.exe     C:\WINDOWS\system32\attrib.exe
2009-05-16 11:58:29 cmd.exe     C:\WINDOWS\system32\attrib.exe
2009-05-16 11:58:35 cmd.exe     E:\SYSTEM\SRC\HS\SBS.EXE
2009-05-16 11:58:44 cmd.exe     E:\SYSTEM\SRC\WIFIKE.EXE
2009-05-16 11:58:47 cmd.exe     E:\SYSTEM\SRC\PWDUMP.EXE
2009-05-16 11:58:48 services.exe    E:\SYSTEM\SRC\imokav.exe
2009-05-16 11:58:50 cmd.exe     E:\SYSTEM\SRC\FGDUMP.EXE
2009-05-16 11:58:52 FGDUMP.EXE  C:\DOCUME~1\test\LOCALS~1\Temp\pwdump.exe
2009-05-16 11:58:54 services.exe    C:\DOCUME~1\test\LOCALS~1\Temp\imokav.exe
2009-05-16 11:58:56 cmd.exe     E:\SYSTEM\SRC\NETPASS.EXE
2009-05-16 11:58:58 cmd.exe     E:\SYSTEM\SRC\MAILPV.EXE
2009-05-16 11:59:01 cmd.exe     E:\SYSTEM\SRC\IEPV.EXE
2009-05-16 11:59:04 cmd.exe     E:\SYSTEM\SRC\MSPASS.EXE
2009-05-16 11:59:06 cmd.exe     E:\SYSTEM\SRC\CACHEDUMP.EXE
2009-05-16 11:59:07 services.exe    E:\SYSTEM\SRC\CACHEDUMP.EXE
2009-05-16 11:59:09 cmd.exe     E:\SYSTEM\SRC\PSPV.EXE
2009-05-16 11:59:11 cmd.exe     E:\SYSTEM\SRC\PRODUKEY.EXE
2009-05-16 11:59:13 cmd.exe     C:\WINDOWS\system32\cscript.exe
2009-05-16 11:59:20 cmd.exe     E:\SYSTEM\SRC\WUL.EXE
```

H.2 Opprettede registernøkler - Dynamisk analyse

```
Dato            Startprogram    øNkkel

2009-05-16 11:57:55 services.exe    HKLM\System\CurrentControlSet\Services\WinVNC\Security
2009-05-16 11:57:55 services.exe    HKLM\System\CurrentControlSet\Services\WinVNC
2009-05-16 11:58:06 regedit.exe HKCU\Software\WinVNC3
2009-05-16 11:58:06 regedit.exe HKCU\Software\ORL\WinVNC3
2009-05-16 11:58:06 regedit.exe HKCU\Software\ORL\VNCHooks\Application_Prefs\WinVNC.exe
2009-05-16 11:58:06 regedit.exe HKCU\Software\ORL\VNCHooks\Application_Prefs
2009-05-16 11:58:06 regedit.exe HKCU\Software\ORL\VNCHooks
2009-05-16 11:58:13 winvnc.exe  HKU\.DEFAULT\Software\ORL
2009-05-16 11:58:06 regedit.exe HKLM\SYSTEM\CurrentControlSet\Services\winvnc\Enum
2009-05-16 11:58:06 regedit.exe HKLM\SYSTEM\CurrentControlSet\Services\winvnc\Security
2009-05-16 11:58:06 regedit.exe HKLM\SYSTEM\CurrentControlSet\Services\winvnc
2009-05-16 11:58:06 regedit.exe HKLM\SOFTWARE\ORL\WinVNC3\Default
2009-05-16 11:58:06 regedit.exe HKLM\SOFTWARE\ORL\WinVNC3
2009-05-16 11:58:13 winvnc.exe  HKLM\SOFTWARE\ORL\WinVNC3\Default
2009-05-16 11:58:13 winvnc.exe  HKLM\Software\ORL\WinVNC3
2009-05-16 11:58:13 winvnc.exe  HKU\.DEFAULT\Software\ORL\VNCHooks\Application_Prefs\winvnc.exe
2009-05-16 11:58:13 winvnc.exe  HKU\.DEFAULT\Software\ORL\VNCHooks\Application_Prefs
2009-05-16 11:58:13 winvnc.exe  HKU\.DEFAULT\Software\ORL\VNCHooks\Application_Prefs\winvnc.exe
2009-05-16 11:58:13 winvnc.exe  HKU\.DEFAULT\Software\ORL\VNCHooks
2009-05-16 11:58:18 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
2009-05-16 11:58:18 LaunchPad.exe   HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
2009-05-16 11:58:18 LaunchPad.exe   HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell
     Folders
2009-05-16 11:58:18 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
2009-05-16 11:58:18 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
2009-05-16 11:58:18 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
2009-05-16 11:58:23 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
2009-05-16 11:58:23 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
2009-05-16 11:58:23 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
2009-05-16 11:58:23 winvnc.exe  HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
...
2009-05-16 11:58:45 WIFIKE.EXE  HKCU\Software\NirSoft
2009-05-16 11:58:45 WIFIKE.EXE  HKCU\Software
2009-05-16 11:58:45 WIFIKE.EXE  HKCU\Software\NirSoft\WirelessKeyView
...
2009-05-16 11:58:47 PWDUMP.EXE  HKLM\SOFTWARE\Microsoft\Cryptography\RNG
```

```
..
2009-05-16 11:58:52 FGDUMP.EXE  HKLM\SOFTWARE\Microsoft\Cryptography\RNG
..
2009-05-16 11:58:57 NETPASS.EXE HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
2009-05-16 11:58:57 NETPASS.EXE HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
2009-05-16 11:58:57 NETPASS.EXE HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
2009-05-16 11:58:57 NETPASS.EXE HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
..
2009-05-16 11:58:57 NETPASS.EXE HKCU\Software\NirSoft\NetPass
..
2009-05-16 11:58:59 MAILPV.EXE  HKLM\SOFTWARE\Microsoft\Cryptography\RNG
..
2009-05-16 11:58:59 MAILPV.EXE  HKCU\Software\NirSoft\MailPassView
..
2009-05-16 11:59:02 IEPV.EXE    HKLM\SOFTWARE\Microsoft\Cryptography\RNG
2009-05-16 11:59:02 IEPV.EXE    HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
2009-05-16 11:59:02 IEPV.EXE    HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
..
2009-05-16 11:59:05 MSPASS.EXE  HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
2009-05-16 11:59:05 MSPASS.EXE  HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
2009-05-16 11:59:05 MSPASS.EXE  HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
2009-05-16 11:59:05 MSPASS.EXE  HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
2009-05-16 11:59:05 MSPASS.EXE  HKCU\Software\NirSoft\MessenPass
..
2009-05-16 11:59:06 CACHEDUMP.EXE   HKLM\SOFTWARE\Microsoft\Cryptography\RNG
```

H.3 Leste filer - Dynamisk analyse

```
Dato            Filnavn     Startprogram

2009-05-16 11:57:25 E:\GO.VBS   wscript.exe
2009-05-16 11:57:25 E:\GO.VBS   wscript.exe
2009-05-16 11:57:25 E:\GO.VBS   wscript.exe
...

2009-05-16 11:57:29 E:\LAUNCHU3.EXE wscript.exe
..
2009-05-16 11:57:30 C:\$Mft LaunchU3.exe
2009-05-16 11:57:33 E:\AUTORUN.INF  LaunchU3.exe
..
2009-05-16 11:57:34 E:\SYSTEM\SRC\GO.BAT    cmd.exe
..
2009-05-16 11:57:34 E:\LaunchPad.zip    LaunchU3.exe
2009-05-16 11:57:34 E:\LaunchPad.zip    LaunchU3.exe
..
2009-05-16 11:57:37 F:\System\SRC\Include\EIP.dat   cmd.exe
2009-05-16 11:57:37 F:\System\SRC\Include\EIP.dat   cmd.exe
..
2009-05-16 11:57:38 C:\WINDOWS\system32\ipconfig.exe    cmd.exe
..
2009-05-16 11:57:38 F:\System\Logs\VICTIM\VICTIM-[20090515-115726].log  cmd.exe
..
2009-05-16 11:57:47 E:\SYSTEM\SRC\WGET.EXE  WGET.EXE
..
2009-05-16 11:57:50 C:\Documents and Settings\test\Application Data\U3\0000060327026187\LaunchPad.exe
     LaunchPad.exe
..
2009-05-16 11:57:52 C:\WINDOWS\system32\xcopy.exe   xcopy.exe
..
2009-05-16 11:58:17 C:\Documents and Settings\test\Application Data\U3\0000060327026187\Launchpad Removal.exe
        LaunchPad.exe
..
2009-05-16 11:58:25 E:\SYSTEM\SRC\HS\OPENSSL.EXE    xcopy.exe
...
2009-05-16 11:58:36 E:\SYSTEM\SRC\HS\SBS.EXE    SBS.EXE
..
2009-05-16 11:58:43 E:\SYSTEM\SRC\WIFIKE.EXE    cmd.exe
..
2009-05-16 11:58:48 E:\SYSTEM\SRC\IMOKAV.EXE    imokav.exe
..
2009-05-16 11:58:51 E:\SYSTEM\SRC\FGDUMP.EXE    FGDUMP.EXE
2009-05-16 11:58:51 \\127.0.0.1\pipe\svcctl FGDUMP.EXE
..
2009-05-16 11:58:56 E:\SYSTEM\SRC\NETPASS.EXE   cmd.exe
..
2009-05-16 11:58:58 E:\SYSTEM\SRC\MAILPV.EXE    cmd.exe
..
2009-05-16 11:59:02 C:\Documents and Settings\test\Local Settings\History\History.IE5\
     MSHist012009050420090505\index.dat    IEPV.EXE
..
2009-05-16 11:59:02 C:\Documents and Settings\test\Local Settings\History\History.IE5\index.dat IEPV.EXE
```

```
..
2009-05-16 11:59:06 E:\SYSTEM\SRC\cachedump.exe cmd.exe
..
2009-05-16 11:59:11 E:\SYSTEM\SRC\PRODUKEY.EXE  cmd.exe
..
2009-05-16 11:59:20 E:\SYSTEM\SRC\WUL.EXE   cmd.exe
```

H.4 Afick logg - overflateanalyse

```
new unknown_type : /mnt/victim_XP/Documents and Settings/test/Application Data/U3
    number of new files      : 15
new unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/History/History.IE5/
     MSHist012009050420090511
    number of new files      : 1
new unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/History/History.IE5/
     MSHist012009051520090516
    number of new files      : 1
new unknown_type : /mnt/victim_XP/WINDOWS/$NtUninstallKB931337$
    number of new files      : 14
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/ATTRIB.EXE-39EAFB02.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/CACHEDUMP.EXE-2003DDD8.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/CSCRIPT.EXE-1C26180C.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/FGDUMP.EXE-24729C4B.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/IEPV.EXE-0CC66364.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/IMOKAV.EXE-16488B1B.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/IMOKAV.EXE-3AD670C3.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/LAUNCHPAD.EXE-29B57FD0.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/LAUNCHU3.EXE-024AD91B.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/MAILPV.EXE-13087CD0.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/MSPASS.EXE-00EB4D42.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/NET.EXE-01A53C2F.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/NET1.EXE-029B9DB4.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/NETPASS.EXE-073AA7BD.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/PRODUKEY.EXE-0AF6A5C7.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/PSPV.EXE-33538890.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/PWDUMP.EXE-0477A877.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/PWDUMP.EXE-242A13CC.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/SBS.EXE-26711756.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/SC.EXE-012262AF.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/TASKMGR.EXE-20256C55.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WGET.EXE-0D89219B.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WIFIKE.EXE-34462AC6.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WINVNC.EXE-2B5AAA4E.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WUL.EXE-0C129EB7.pf
new unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/XCOPY.EXE-21FC761A.pf
new unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/Logs/tmp.edb
new unknown_type : /mnt/victim_XP/WINDOWS/Temp/Perflib_Perfdata_7d4.dat
new unknown_type : /mnt/victim_XP/WINDOWS/VNCHOOKS.DLL
new unknown_type : /mnt/victim_XP/WINDOWS/WINVNC.EXE
new unknown_type : /mnt/victim_XP/agent/log.csv
new unknown_type : /mnt/victim_XP/agent/log.pml
deleted unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/History/History.IE5/
     MSHist012009050420090505
    number of deleted files      : 1
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/Application Data/IconCache.
     db
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/History/History.IE5/index.
     dat
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/Local Settings/Temporary Internet Files/
     Content.IE5/index.dat
changed unknown_type : /mnt/victim_XP/Documents and Settings/test/NTUSER.DAT
changed unknown_type : /mnt/victim_XP/System Volume Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB
     }/_driver.cfg
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/CMD.EXE-087B4001.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/IMAPI.EXE-0BF740A4.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/IPCONFIG.EXE-2395F30B.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/PROCMON.EXE-09DDAFE0.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/REG.EXE-0D2A95F7.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/REGEDIT.EXE-1B606482.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/RUNDLL32.EXE-451FC2C0.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/VMIP.EXE-2BD5723A.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WMIPRVSE.EXE-28F301A9.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WSCNTFY.EXE-1B24F5EB.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/Prefetch/WSCRIPT.EXE-32960AB9.pf
changed unknown_type : /mnt/victim_XP/WINDOWS/SchedLgU.Txt
changed unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/DataStore.edb
changed unknown_type : /mnt/victim_XP/WINDOWS/SoftwareDistribution/DataStore/Logs/edb.chk
changed unknown_type : /mnt/victim_XP/WINDOWS/Tasks/SA.DAT
changed unknown_type : /mnt/victim_XP/WINDOWS/bootstat.dat
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/CatRoot2/edb.chk
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/CatRoot2/{F750E6C3-38EE-11D1-85E5-00C04FC295EE}/catdb
```

```
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/AppEvent.Evt
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/SAM
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/SysEvent.Evt
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/default
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/software
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/config/system
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/INDEX.BTR
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/INDEX.MAP
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/MAPPING1.MAP
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/MAPPING2.MAP
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/OBJECTS.DATA
changed unknown_type : /mnt/victim_XP/WINDOWS/system32/wbem/Repository/FS/OBJECTS.MAP
changed unknown_type : /mnt/victim_XP/pagefile.sys

# Hash database : 9253 files scanned, 105 changed (new : 67; delete : 2; changed : 36; dangling : 0;
     exclude_suffix : 1803; exclude_prefix : 0; exclude_re : 0; degraded : 12)
# ##################################################################
# MD5 hash of /var/lib/afick/afick => yhx4J1+WnKdvt5Qxb/FhQw
# user time : 17.04; system time : 10.92; real time : 268
```

H.5 MACtimes - overflateanalyse

```
Fri May 15 2009 11:35:00       93 m.. -r-------- 0        0        11055    /mnt/victim_XP/WINDOWS/
     $NtUninstallKB931337$/stunnel.conf
Fri May 15 2009 11:56:45   303104 .a. dr-x------ 0        0        29       /mnt/victim_XP/WINDOWS/system32
Fri May 15 2009 11:56:46   708096 .a. -r-------- 0        0        2060     /mnt/victim_XP/WINDOWS/system32/
     ntdll.dll
                           474112 .a. -r-------- 0        0        2080     /mnt/victim_XP/WINDOWS/system32/
                                shlwapi.dll
                           218624 .a. -r-------- 0        0        2156     /mnt/victim_XP/WINDOWS/system32/
                                uxtheme.dll
                                0 .a. dr-x------ 0        0        10498    /mnt/victim_XP/Program Files/
                                    VMware
                           983552 .a. -r-------- 0        0        2069     /mnt/victim_XP/WINDOWS/system32/
                                kernel32.dll
                           278016 .a. -r-------- 0        0        2067     /mnt/victim_XP/WINDOWS/system32/
                                gdi32.dll
                                0 .a. dr-x------ 0        0        3568     /mnt/victim_XP/WINDOWS/WinSxS/
                                    x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-
                                    ww_a84f1ff9
                                0 .a. dr-x------ 0        0        10539    /mnt/victim_XP/Documents and
                                    Settings/All Users/Application Data/VMware/VMware Tools
                           611328 .a. -r-------- 0        0        2081     /mnt/victim_XP/WINDOWS/system32/
                                comctl32.dll
                           176128 .a. -r-------- 0        0        2137     /mnt/victim_XP/WINDOWS/system32/
                                winmm.dll
                             8192 .a. dr-x------ 0        0        10499    /mnt/victim_XP/Program Files/
                                  VMware/VMware Tools
                             8192 .a. dr-x------ 0        0        126      /mnt/victim_XP/WINDOWS/WinSxS
                           577024 .a. -r-------- 0        0        2076     /mnt/victim_XP/WINDOWS/system32/
                                user32.dll
                            71680 .a. -r-------- 0        0        2165     /mnt/victim_XP/WINDOWS/system32/
                                 msacm32.dll
                            65536 .a. -r-------- 0        0        2973     /mnt/victim_XP/WINDOWS/system32/
                                 shimeng.dll
                           126976 .a. -r-------- 0        0        2119     /mnt/victim_XP/WINDOWS/system32/
                                apphelp.dll
                           343040 .a. -r-------- 0        0        2082     /mnt/victim_XP/WINDOWS/system32/
                                msvcrt.dll
                          1050624 .a. -r-------- 0        0        3569     /mnt/victim_XP/WINDOWS/WinSxS/
                               x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-
                               ww_a84f1ff9/comctl32.dll
                           616960 .a. -r-------- 0        0        2065     /mnt/victim_XP/WINDOWS/system32/
                                advapi32.dll
                           723456 .a. -r-------- 0        0        2087     /mnt/victim_XP/WINDOWS/system32/
                                userenv.dll
                           553472 .a. -r-------- 0        0        2071     /mnt/victim_XP/WINDOWS/system32/
                                oleaut32.dll
                          1281536 .a. -r-------- 0        0        2070     /mnt/victim_XP/WINDOWS/system32/
                               ole32.dll
                          1852416 .a. -r-------- 0        0        2313     /mnt/victim_XP/WINDOWS/AppPatch/
                               AcGenral.dll
                             4096 .a. dr-x------ 0        0        3989     /mnt/victim_XP/Program Files
                            18944 .a. -r-------- 0        0        2077     /mnt/victim_XP/WINDOWS/system32/
                                 version.dll
                             1939 .a. -r-------- 0        0        10495    /mnt/victim_XP/Documents and
                                  Settings/All Users/Application Data/VMware/VMware Tools/manifest.txt
                           581120 .a. -r-------- 0        0        2072     /mnt/victim_XP/WINDOWS/system32/
                                rpcrt4.dll
                           110080 .a. -r-------- 0        0        2136     /mnt/victim_XP/WINDOWS/system32/
                                imm32.dll
```

```
                          4096 .a. dr-x------ 0        0        87       /mnt/victim_XP/WINDOWS/AppPatch
                        294400 .a. -r-------- 0        0        3176     /mnt/victim_XP/WINDOWS/system32/
                             MSCTF.dll
                             0 .a. dr-x------ 0        0        5776     /mnt/victim_XP/WINDOWS/Tasks
Fri May 15 2009 11:56:47     0 .a. dr-x------ 0        0        10574    /mnt/victim_XP/WINDOWS/WinSxS/
     x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_6b128700
                         94720 .a. -r-------- 0        0        2150     /mnt/victim_XP/WINDOWS/system32/
                              iphlpapi.dll
                         25088 .a. -r-------- 0        0        2202     /mnt/victim_XP/WINDOWS/system32/
                              shfolder.dll
                             0 .a. dr-x------ 0        0        3600     /mnt/victim_XP/Documents and
                                 Settings/All Users/Application Data
                          1869 .a. -r-------- 0        0        10572    /mnt/victim_XP/WINDOWS/WinSxS/
                               Manifests/x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.762_x-
                               ww_6b128700.manifest
                          4096 .a. dr-x------ 0        0        10578    /mnt/victim_XP/WINDOWS/WinSxS/
                               Policies/x86_policy.8.0.Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_x-ww_77c24773
                        626688 .a. -r-------- 0        0        10575    /mnt/victim_XP/WINDOWS/WinSxS/
                             x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_6b128700/msvcr80.
                             dll
                             0 .a. dr-x------ 0        0        10829    /mnt/victim_XP/Documents and
                                 Settings/test/Local Settings/Temp/VMwareDnD/05806cc4
                        322096 .a. -r-------- 0        0        10554    /mnt/victim_XP/Program Files/
                             VMware/VMware Tools/VMip.exe
                          4096 .a. dr-x------ 0        0        3520     /mnt/victim_XP/Documents and
                               Settings/All Users
                             0 .a. dr-x------ 0        0        10496    /mnt/victim_XP/Documents and
                                 Settings/test/Local Settings/Temp/VMwareDnD
                           409 .a. -r-------- 0        0        10828    /mnt/victim_XP/Documents and
                                Settings/All Users/Application Data/VMware/VMware Tools/tools.conf
                           621 .a. -r-------- 0        0        3574     /mnt/victim_XP/WINDOWS/WinSxS/
                                Policies/x86_policy.6.0.Microsoft.Windows.Common-
                                Controls_6595b64144ccf1df_x-ww_5ddad775/6.0.2600.2180.Policy
                         19968 .a. -r-------- 0        0        2106     /mnt/victim_XP/WINDOWS/system32/
                              ws2help.dll
                          4096 .a. dr-x------ 0        0        3572     /mnt/victim_XP/WINDOWS/WinSxS/
                               Policies/x86_policy.6.0.Microsoft.Windows.Common-
                               Controls_6595b64144ccf1df_x-ww_5ddad775
                           338 .a. -r-------- 0        0        10550    /mnt/victim_XP/Program Files/
                                VMware/VMware Tools/resume-vm-default.bat
                         55808 .a. -r-------- 0        0        2100     /mnt/victim_XP/WINDOWS/system32/
                              secur32.dll
                             0 .a. dr-x------ 0        0        10538    /mnt/victim_XP/Documents and
                                 Settings/All Users/Application Data/VMware
                        148480 .a. -r-------- 0        0        2129     /mnt/victim_XP/WINDOWS/system32/
                             dnsapi.dll
                          3584 .a. -r-------- 0        0        2193     /mnt/victim_XP/WINDOWS/system32/
                               icmp.dll
                          1862 .a. -r-------- 0        0        3571     /mnt/victim_XP/WINDOWS/WinSxS/
                               Manifests/x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6
                               .0.2600.2180_x-ww_a84f1ff9.Manifest
                        111104 .a. -r-------- 0        0        2151     /mnt/victim_XP/WINDOWS/system32/
                             dhcpcsvc.dll
                         82944 .a. -r-------- 0        0        2105     /mnt/victim_XP/WINDOWS/system32/
                              ws2_32.dll
                           800 .a. -r-------- 0        0        10579    /mnt/victim_XP/WINDOWS/WinSxS/
                                Policies/x86_policy.8.0.Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_x-ww_77c24773
                                /8.0.50727.762.policy
                        132608 .a. -r-------- 0        0        3080     /mnt/victim_XP/WINDOWS/system32/
                             upnp.dll
                           749 .a. -r-------- 0        0        6402     /mnt/victim_XP/WINDOWS/
                                WindowsShell.Manifest
                          4096 .a. dr-x------ 0        0        3518     /mnt/victim_XP/Documents and
                               Settings
Fri May 15 2009 11:56:48    12458 mac -r-------- 0        0        7399     /mnt/victim_XP/WINDOWS/Prefetch/
     VMIP.EXE-2BD5723A.pf
Fri May 15 2009 11:56:51    16896 .a. -r-------- 0        0        5610     /mnt/victim_XP/WINDOWS/system32/
     fltlib.dll
                         23040 .a. -r-------- 0        0        2271     /mnt/victim_XP/WINDOWS/system32/
                              psapi.dll
                       83717408 .a. -r-------- 0        0        11003    /mnt/victim_XP/agent/log.pml
                          28672 m.c dr-x------ 0        0        31       /mnt/victim_XP/WINDOWS/system32/
                              drivers
                        2902376 .a. -r-------- 0        0        10930    /mnt/victim_XP/agent/Procmon.exe
                         290816 .a. -r-------- 0        0        2094     /mnt/victim_XP/WINDOWS/system32/
                             winsrv.dll
                         640000 .a. -r-------- 0        0        2289     /mnt/victim_XP/WINDOWS/system32/
                             dbghelp.dll
                         276992 .a. -r-------- 0        0        2066     /mnt/victim_XP/WINDOWS/system32/
                             comdlg32.dll
                         395776 .a. -r-------- 0        0        2146     /mnt/victim_XP/WINDOWS/system32/
                             rpcss.dll
                           252 .a. -r-------- 0        0        10932    /mnt/victim_XP/agent/start.bat
```

```
Fri May 15 2009 11:56:52   472064 .a. -r-------- 0        0        4725     /mnt/victim_XP/WINDOWS/system32/
     wbem/fastprox.dll
                            43520 .a. -r-------- 0        0        4751     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/wbemsvc.dll
                                0 .a. dr-x------ 0        0        10500    /mnt/victim_XP/Program Files/
                                     VMware/VMware Tools/Drivers
                           218112 .a. -r-------- 0        0        4767     /mnt/victim_XP/WINDOWS/system32/
                                wbem/wmiprvse.exe
                                0 .a. dr-x------ 0        0        62       /mnt/victim_XP/WINDOWS/system32/
                                     wbem/Repository
                             4096 .a. dr-x------ 0        0        32       /mnt/victim_XP/WINDOWS/system
                            18944 .a. -r-------- 0        0        4750     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/wbemprox.dll
                            67072 .a. -r-------- 0        0        2131     /mnt/victim_XP/WINDOWS/system32/
                                 ntdsapi.dll
                                0 .a. dr-x------ 0        0        3695     /mnt/victim_XP/Documents and
                                     Settings/Default User/Start Menu
                           792064 .a. -r-------- 0        0        2405     /mnt/victim_XP/WINDOWS/system32/
                                comres.dll
                             4096 .a. dr-x------ 0        0        3698     /mnt/victim_XP/Documents and
                                  Settings/Default User/Start Menu/Programs
                            28672 .a. dr-x------ 0        0        31       /mnt/victim_XP/WINDOWS/system32/
                                 drivers
                           413696 .a. -r-------- 0        0        2189     /mnt/victim_XP/WINDOWS/system32/
                                msvcp60.dll
                            36864 .a. dr-x------ 0        0        61       /mnt/victim_XP/WINDOWS/system32/
                                 wbem
                                0 .a. dr-x------ 0        0        10607    /mnt/victim_XP/Program Files/
                                     VMware/VMware Tools/Drivers/memctl
                            49152 .a. dr-x------ 0        0        3519     /mnt/victim_XP/Documents and
                                  Settings/Default User
                          2897920 .a. -r-------- 0        0        3226     /mnt/victim_XP/WINDOWS/system32/
                               xpsp2res.dll
                             4096 .a. dr-x------ 0        0        5395     /mnt/victim_XP/WINDOWS/system32/
                                  wbem/Repository/FS
                           501248 .a. -r-------- 0        0        4781     /mnt/victim_XP/WINDOWS/system32/
                                clbcatq.dll
                             4096 .a. dr-x------ 0        0        6386     /mnt/victim_XP/Documents and
                                  Settings/Default User/Start Menu/Programs/Accessories
                           214528 .a. -r-------- 0        0        4745     /mnt/victim_XP/WINDOWS/system32/
                                wbem/wbemcomn.dll
                           502272 .a. -r-------- 0        0        2098     /mnt/victim_XP/WINDOWS/system32/
                                winlogon.exe
Fri May 15 2009 11:56:53     4096 .a. dr-x------ 0        0        9997     /mnt/victim_XP/WINDOWS/system32/
     config/systemprofile/Start Menu/Programs
                             4096 .a. dr-x------ 0        0        144      /mnt/victim_XP/WINDOWS/system32/
                                  wbem/Logs
                                0 .a. dr-x------ 0        0        43       /mnt/victim_XP/WINDOWS/system32/
                                     drivers/etc
                             4096 .a. dr-x------ 0        0        9999     /mnt/victim_XP/WINDOWS/system32/
                                  config/systemprofile/Start Menu/Programs/Accessories
                             4096 .a. dr-x------ 0        0        30       /mnt/victim_XP/WINDOWS/system32/
                                  config
                             4096 .a. dr-x------ 0        0        42       /mnt/victim_XP/WINDOWS/repair
                             4096 .a. dr-x------ 0        0        9994     /mnt/victim_XP/WINDOWS/system32/
                                  config/systemprofile
                              734 .a. -r-------- 0        0        236      /mnt/victim_XP/WINDOWS/system32/
                                   drivers/etc/hosts
                             4096 .a. dr-x------ 0        0        5351     /mnt/victim_XP/WINDOWS/
                                  Registration
                                0 .a. dr-x------ 0        0        9996     /mnt/victim_XP/WINDOWS/system32/
                                     config/systemprofile/Start Menu
Fri May 15 2009 11:56:54    14336 .a. -r-------- 0        0        2145     /mnt/victim_XP/WINDOWS/system32/
     svchost.exe
                            18432 .a. -r-------- 0        0        2157     /mnt/victim_XP/WINDOWS/system32/
                                 wtsapi32.dll
                          1352192 .a. -r-------- 0        0        4722     /mnt/victim_XP/WINDOWS/system32/
                               wbem/cimwin32.dll
                           172032 .a. -r-------- 0        0        2079     /mnt/victim_XP/WINDOWS/system32/
                                wldap32.dll
                             5632 .a. -r-------- 0        0        2959     /mnt/victim_XP/WINDOWS/system32/
                                  security.dll
                           108032 .a. -r-------- 0        0        2121     /mnt/victim_XP/WINDOWS/system32/
                                services.exe
                           144896 .a. -r-------- 0        0        2132     /mnt/victim_XP/WINDOWS/system32/
                                schannel.dll
                           597504 .a. -r-------- 0        0        2086     /mnt/victim_XP/WINDOWS/system32/
                                crypt32.dll
                            31232 .a. -r-------- 0        0        1500     /mnt/victim_XP/WINDOWS/system32/
                                 traffic.dll
                            95232 .a. -r-------- 0        0        4770     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/wmiutils.dll
                           185856 .a. -r-------- 0        0        4726     /mnt/victim_XP/WINDOWS/system32/
```

```
                            wbem/framedyn.dll
                         983552 .a. -r-------- 0        0        2115     /mnt/victim_XP/WINDOWS/system32/
                            setupapi.dll
                         332288 .a. -r-------- 0        0        2103     /mnt/victim_XP/WINDOWS/system32/
                            netapi32.dll
                          53760 .a. -r-------- 0        0        2101     /mnt/victim_XP/WINDOWS/system32/
                            winsta.dll
                          36352 .a. -r-------- 0        0        2127     /mnt/victim_XP/WINDOWS/system32/
                            ncobjapi.dll
                          16896 .a. -r-------- 0        0        2167     /mnt/victim_XP/WINDOWS/system32/
                            cfgmgr32.dll
                           6144 .a. -r-------- 0        0        2091     /mnt/victim_XP/WINDOWS/system32/
                            csrss.exe
                         144896 .a. -r-------- 0        0        4765     /mnt/victim_XP/WINDOWS/system32/
                            wbem/wmiprov.dll
                          92672 .a. -r-------- 0        0        2480     /mnt/victim_XP/WINDOWS/system32/
                            dskquota.dll
                          50688 .a. -r-------- 0        0        2061     /mnt/victim_XP/WINDOWS/system32/
                            smss.exe
                           5632 .a. -r-------- 0        0        2273     /mnt/victim_XP/WINDOWS/system32/
                            wmi.dll
                          57344 .a. -r-------- 0        0        2088     /mnt/victim_XP/WINDOWS/system32/
                            msasn1.dll
Fri May 15 2009 11:56:55    13312 .a. -r-------- 0        0        2122     /mnt/victim_XP/WINDOWS/system32/
     lsass.exe
                           4096 .a. dr-x------ 0        0        141      /mnt/victim_XP/WINDOWS/pchealth/
                            helpctr/binaries
                           4096 .a. dr-x------ 0        0        140      /mnt/victim_XP/WINDOWS/pchealth/
                            helpctr
                              0 .a. dr-x------ 0        0        139      /mnt/victim_XP/WINDOWS/pchealth
Fri May 15 2009 11:56:56  2902376 ..c -r-------- 0        0        10930    /mnt/victim_XP/agent/Procmon.exe
                            602 mac -r-------- 0        0        5393     /mnt/victim_XP/WINDOWS/system32/
                             wbem/Logs/wmiprov.log
                         346672 .a. -r-------- 0        0        10712    /mnt/victim_XP/Program Files/
                            VMware/VMware Tools/vmacthlp.exe
                           3196 mac -r-------- 0        0        5398     /mnt/victim_XP/WINDOWS/system32/
                            wbem/Repository/FS/MAPPING2.MAP
Fri May 15 2009 11:56:57       0 .a. dr-x------ 0        0        38       /mnt/victim_XP/WINDOWS/system32/
     spool/prtprocs
                              0 .a. dr-x------ 0        0        34       /mnt/victim_XP/WINDOWS/system32/
                                 spool
                          19968 .a. -r-------- 0        0        2148     /mnt/victim_XP/WINDOWS/system32/
                            wshtcpip.dll
                           7116 .a. -r-------- 0        0        1397     /mnt/victim_XP/WINDOWS/system32/
                            drivers/etc/services
                          57856 .a. -r-------- 0        0        2162     /mnt/victim_XP/WINDOWS/system32/
                            spoolsv.exe
                        1032192 .a. -r-------- 0        0        2218     /mnt/victim_XP/WINDOWS/explorer.
                             exe
                         344064 .a. -r-------- 0        0        2278     /mnt/victim_XP/WINDOWS/system32/
                            hnetcfg.dll
                          16896 .a. -r-------- 0        0        2152     /mnt/victim_XP/WINDOWS/system32/
                            winrnr.dll
                              0 .a. dr-x------ 0        0        39       /mnt/victim_XP/WINDOWS/system32/
                                 spool/prtprocs/w32x86
                           8192 .a. -r-------- 0        0        2153     /mnt/victim_XP/WINDOWS/system32/
                            rasadhlp.dll
                         245248 .a. -r-------- 0        0        2147     /mnt/victim_XP/WINDOWS/system32/
                            mswsock.dll
Fri May 15 2009 11:56:58       0 .a. dr-x------ 0        0        10583    /mnt/victim_XP/WINDOWS/WinSxS/
     x86_Microsoft.VC80.MFC_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_3bf8fa05
                         416304 .a. -r-------- 0        0        10559    /mnt/victim_XP/Program Files/
                            VMware/VMware Tools/VMwareTray.exe
                           4096 .a. dr-x------ 0        0        10593    /mnt/victim_XP/WINDOWS/WinSxS/
                             x86_Microsoft.VC80.MFCLOC_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_91481303
                         111104 .a. -r-------- 0        0        5693     /mnt/victim_XP/WINDOWS/system32/
                            wuauclt.exe
                          32256 .a. -r-------- 0        0        3118     /mnt/victim_XP/WINDOWS/system32/
                            wpabaln.exe
                          13824 .a. -r-------- 0        0        3121     /mnt/victim_XP/WINDOWS/system32/
                            wscntfy.exe
                          44544 .a. -r-------- 0        0        2335     /mnt/victim_XP/WINDOWS/system32/
                            alg.exe
                          69120 .a. -r-------- 0        0        2288     /mnt/victim_XP/WINDOWS/system32/
                            notepad.exe
                         539184 .a. -r-------- 0        0        10558    /mnt/victim_XP/Program Files/
                            VMware/VMware Tools/VMwareService.exe
                         830000 .a. -r-------- 0        0        10560    /mnt/victim_XP/Program Files/
                            VMware/VMware Tools/VMwareUser.exe
Fri May 15 2009 11:56:59    15360 .a. -r-------- 0        0        3181     /mnt/victim_XP/WINDOWS/system32/
     ctfmon.exe
Fri May 15 2009 11:57:04  1114112 mac -r-------- 0        0        5403     /mnt/victim_XP/WINDOWS/system32/
     wbem/Repository/FS/INDEX.BTR
```

```
                             568 mac -r-------- 0        0        5400     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/INDEX.MAP
                         5332992 mac -r-------- 0        0        5402     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/OBJECTS.DATA
                            2628 mac -r-------- 0        0        5401     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/OBJECTS.MAP
                            4096 .a. dr-x------ 0        0        10256    /mnt/victim_XP/Documents and
                                 Settings/test/Local Settings
                               4 mac -r-------- 0        0        5399     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/MAPPING.VER
                            3196 mac -r-------- 0        0        5397     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/Repository/FS/MAPPING1.MAP
Fri May 15 2009 11:57:05    31276 mac -r-------- 0        0        10226    /mnt/victim_XP/WINDOWS/Prefetch/
     WMIPRVSE.EXE-28F301A9.pf
Fri May 15 2009 11:57:12    26496 .a. -r-------- 0        0        10838    /mnt/victim_XP/WINDOWS/system32/
     drivers/USBSTOR.SYS
                            41856 .a. -r-------- 0        0        2059     /mnt/victim_XP/WINDOWS/system32/
                                 drivers/imapi.sys
Fri May 15 2009 11:57:13    33280 .a. -r-------- 0        0        2248     /mnt/victim_XP/WINDOWS/system32/
     rundll32.exe
                           150016 .a. -r-------- 0        0        2251     /mnt/victim_XP/WINDOWS/system32/
                                 imapi.exe
                           144384 .a. -r-------- 0        0        2068     /mnt/victim_XP/WINDOWS/system32/
                                 imagehlp.dll
Fri May 15 2009 11:57:14   143360 .a. -r-------- 0        0        2063     /mnt/victim_XP/WINDOWS/system32/
     drivers/fastfat.sys
Fri May 15 2009 11:57:16   176640 .a. -r-------- 0        0        2117     /mnt/victim_XP/WINDOWS/system32/
     wintrust.dll
                           101888 .a. -r-------- 0        0        2238     /mnt/victim_XP/WINDOWS/system32/
                                 actxprxy.dll
                           152576 .a. -r-------- 0        0        2141     /mnt/victim_XP/WINDOWS/system32/
                                 rsaenh.dll
Fri May 15 2009 11:57:18     4096 .a. dr-x------ 0        0        10131    /mnt/victim_XP/Documents and
     Settings/test
                                0 .a. dr-x------ 0        0        3708     /mnt/victim_XP/Documents and
                                     Settings/All Users/Desktop
                                0 .a. dr-x------ 0        0        10272    /mnt/victim_XP/Documents and
                                     Settings/test/Desktop
                            71680 .a. -r-------- 0        0        4746     /mnt/victim_XP/WINDOWS/system32/
                                 wbem/wbemcons.dll
Fri May 15 2009 11:57:20       75 .a. -r-------- 0        0        10345    /mnt/victim_XP/Documents and
     Settings/test/My Documents/desktop.ini
                           114688 ..c -r-------- 0        0        3122     /mnt/victim_XP/WINDOWS/system32/
                                 wscript.exe
                             4096 .a. dr-x------ 0        0        10255    /mnt/victim_XP/Documents and
                                  Settings/test/My Documents
                             4096 mac dr-x------ 0        0        72       /mnt/victim_XP/WINDOWS/Temp
                            11976 mac -r-------- 0        0        10484    /mnt/victim_XP/WINDOWS/Prefetch/
                                  RUNDLL32.EXE-451FC2C0.pf
Fri May 15 2009 11:57:21   114688 .a. -r-------- 0        0        3122     /mnt/victim_XP/WINDOWS/system32/
     wscript.exe
                                0 .a. dr-x------ 0        0        93       /mnt/victim_XP/WINDOWS/Resources/
                                     Themes
                           361472 .a. -r-------- 0        0        357      /mnt/victim_XP/WINDOWS/Resources/
                                 Themes/Luna/Shell/NormalColor/shellstyle.dll
                                0 .a. dr-x------ 0        0        94       /mnt/victim_XP/WINDOWS/Resources/
                                     Themes/Luna
                                0 .a. dr-x------ 0        0        92       /mnt/victim_XP/WINDOWS/Resources
Fri May 15 2009 11:57:24    59904 .a. -r-------- 0        0        2083     /mnt/victim_XP/WINDOWS/system32/
     mpr.dll
                            44032 .a. -r-------- 0        0        2744     /mnt/victim_XP/WINDOWS/system32/
                                  msisip.dll
                           146432 .a. -r-------- 0        0        2210     /mnt/victim_XP/WINDOWS/system32/
                                 winspool.drv
                           159744 .a. -r-------- 0        0        2945     /mnt/victim_XP/WINDOWS/system32/
                                 scrobj.dll
                          1028096 .a. -r-------- 0        0        2686     /mnt/victim_XP/WINDOWS/system32/
                                 mfc42.dll
                           151552 .a. -r-------- 0        0        2946     /mnt/victim_XP/WINDOWS/system32/
                                 scrrun.dll
                           713216 .a. -r-------- 0        0        2112     /mnt/victim_XP/WINDOWS/system32/
                                 sxs.dll
                            65536 .a. -r-------- 0        0        3128     /mnt/victim_XP/WINDOWS/system32/
                                  wshext.dll
                            98304 .a. -r-------- 0        0        3130     /mnt/victim_XP/WINDOWS/system32/
                                  wshom.ocx
                           417792 .a. -r-------- 0        0        3094     /mnt/victim_XP/WINDOWS/system32/
                                 vbscript.dll
Fri May 15 2009 11:57:26    16506 mac -r-------- 0        0        10896    /mnt/victim_XP/WINDOWS/Prefetch/
     IMAPI.EXE-0BF740A4.pf
Fri May 15 2009 11:57:27  1492480 .a. -r-------- 0        0        2221     /mnt/victim_XP/WINDOWS/system32/
     shdocvw.dll
                           656384 .a. -r-------- 0        0        2078     /mnt/victim_XP/WINDOWS/system32/
```

```
                                wininet.dll
                           431616 .a. -r-------- 0        0        2269     /mnt/victim_XP/WINDOWS/system32/
                                riched20.dll
Fri May 15 2009 11:57:28   612352 .a. -r-------- 0        0        2075     /mnt/victim_XP/WINDOWS/system32/
     urlmon.dll
                           101888 .a. -r-------- 0        0        2208     /mnt/victim_XP/WINDOWS/system32/
                                cscdll.dll
                           326656 .a. -r-------- 0        0        2211     /mnt/victim_XP/WINDOWS/system32/
                                cscui.dll
Fri May 15 2009 11:57:29        0 .a. dr-x------ 0        0        10413    /mnt/victim_XP/Documents and
     Settings/test/Local Settings/Application Data/Microsoft/CD Burning
Fri May 15 2009 11:57:30    20992 .a. -r-------- 0        0        2548     /mnt/victim_XP/WINDOWS/system32/
     hid.dll
Fri May 15 2009 11:57:31        0 .a. dr-x------ 0        0        10152    /mnt/victim_XP/Documents and
     Settings/test/Start Menu
                             4096 .a. dr-x------ 0        0        3711     /mnt/victim_XP/Documents and
                                  Settings/All Users/Start Menu/Programs
                             4096 .a. dr-x------ 0        0        10236    /mnt/victim_XP/Documents and
                                  Settings/test/Start Menu/Programs
                             4096 .a. dr-x------ 0        0        3709     /mnt/victim_XP/Documents and
                                  Settings/All Users/Start Menu
Fri May 15 2009 11:57:33     4096 mac dr-x------ 0        0        10274    /mnt/victim_XP/Documents and
     Settings/test/Application Data
Fri May 15 2009 11:57:34       12 .ac -r-------- 0        0        11007    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/version.dat
Fri May 15 2009 11:57:37   110592 .ac -r-------- 0        0        11008    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/cleanup.exe
Fri May 15 2009 11:57:38    55808 .a. -r-------- 0        0        2252     /mnt/victim_XP/WINDOWS/system32/
     ipconfig.exe
                            26106 mac -r-------- 0        0        10842    /mnt/victim_XP/WINDOWS/Prefetch/
                                  WSCRIPT.EXE-32960AB9.pf
Fri May 15 2009 11:57:39  3096576 .ac -r-------- 0        0        11009    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/Launchpad Removal.exe
Fri May 15 2009 11:57:41  1082368 .a. -r-------- 0        0        2180     /mnt/victim_XP/WINDOWS/system32/
     esent.dll
                            58880 .a. -r-------- 0        0        2179     /mnt/victim_XP/WINDOWS/system32/
                                  atl.dll
                           163840 .a. -r-------- 0        0        2281     /mnt/victim_XP/WINDOWS/system32/
                                credui.dll
                           198144 .a. -r-------- 0        0        2253     /mnt/victim_XP/WINDOWS/system32/
                                netman.dll
                           359936 .a. -r-------- 0        0        3139     /mnt/victim_XP/WINDOWS/system32/
                                wzcsvc.dll
                           181760 .a. -r-------- 0        0        2206     /mnt/victim_XP/WINDOWS/system32/
                                tapi32.dll
                            61440 .a. -r-------- 0        0        2205     /mnt/victim_XP/WINDOWS/system32/
                                  rasman.dll
                            64000 .a. -r-------- 0        0        2130     /mnt/victim_XP/WINDOWS/system32/
                                  samlib.dll
                            13300 mac -r-------- 0        0        11011    /mnt/victim_XP/WINDOWS/Prefetch/
                                  LAUNCHU3.EXE-024AD91B.pf
                           194048 .a. -r-------- 0        0        2195     /mnt/victim_XP/WINDOWS/system32/
                                activeds.dll
                           236544 .a. -r-------- 0        0        2204     /mnt/victim_XP/WINDOWS/system32/
                                rasapi32.dll
                            51712 .a. -r-------- 0        0        3138     /mnt/victim_XP/WINDOWS/system32/
                                  wzcsapi.dll
                            87040 .a. -r-------- 0        0        2194     /mnt/victim_XP/WINDOWS/system32/
                                  mprapi.dll
                            44032 .a. -r-------- 0        0        2155     /mnt/victim_XP/WINDOWS/system32/
                                  rtutils.dll
                           143360 .a. -r-------- 0        0        2196     /mnt/victim_XP/WINDOWS/system32/
                                adsldpc.dll
                          1708032 .a. -r-------- 0        0        2274     /mnt/victim_XP/WINDOWS/system32/
                                netshell.dll
Fri May 15 2009 11:57:42  4608000 .ac -r-------- 0        0        11010    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/LaunchPad.exe
                              328 .ac -r-------- 0        0        11013    /mnt/victim_XP/Documents and
                                    Settings/test/Application Data/U3/0000060327026187/Loading.htm
                            58842 .ac -r-------- 0        0        11012    /mnt/victim_XP/Documents and
                                  Settings/test/Application Data/U3/0000060327026187/Loading.gif
Fri May 15 2009 11:57:43   109621 .ac -r-------- 0        0        11016    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/LPHelp-en.chm
                            98339 .ac -r-------- 0        0        11015    /mnt/victim_XP/Documents and
                                  Settings/test/Application Data/U3/0000060327026187/LPHelp-de.chm
                            78576 .ac -r-------- 0        0        11014    /mnt/victim_XP/Documents and
                                  Settings/test/Application Data/U3/0000060327026187/LPHelp-ch.chm
Fri May 15 2009 11:57:44    90017 .ac -r-------- 0        0        11020    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/LPHelp-jp.chm
                            95968 .ac -r-------- 0        0        11018    /mnt/victim_XP/Documents and
                                  Settings/test/Application Data/U3/0000060327026187/LPHelp-fr.chm
                            94331 .ac -r-------- 0        0        11019    /mnt/victim_XP/Documents and
                                  Settings/test/Application Data/U3/0000060327026187/LPHelp-it.chm
```

```
                           94194 .ac -r-------- 0        0        11017    /mnt/victim_XP/Documents and
                                 Settings/test/Application Data/U3/0000060327026187/LPHelp-es.chm
Fri May 15 2009 11:57:45   528384 .ac -r-------- 0        0        11021    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/SanDiskFormatExtension.dll
                            1901 .ac -r-------- 0        0        11022    /mnt/victim_XP/Documents and
                                  Settings/test/Application Data/U3/0000060327026187/SanDiskFormatExtension.
                                  dll.sig
Fri May 15 2009 11:57:46     1901 .ac -r-------- 0        0        11024    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/SanDiskSecurityExtension.dll.sig
                           49152 .ac -r-------- 0        0        11025    /mnt/victim_XP/Documents and
                                 Settings/test/Application Data/U3/0000060327026187/U3AccessGrant.exe
                         2260992 .ac -r-------- 0        0        11023    /mnt/victim_XP/Documents and
                                Settings/test/Application Data/U3/0000060327026187/SanDiskSecurityExtension.
                                dll
                           19244 mac -r-------- 0        0        10447    /mnt/victim_XP/WINDOWS/Prefetch/
                                  IPCONFIG.EXE-2395F30B.pf
Fri May 15 2009 11:57:47   622592 .a. -r-------- 0        0        11027    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/U3LauncherSetup.msi
                         1163264 .ac -r-------- 0        0        11026    /mnt/victim_XP/Documents and
                                 Settings/test/Application Data/U3/0000060327026187/u3dapi10.dll
                           22528 .a. -r-------- 0        0        2192     /mnt/victim_XP/WINDOWS/system32/
                                  wsock32.dll
Fri May 15 2009 11:57:48   622592 ..c -r-------- 0        0        11027    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/0000060327026187/U3LauncherSetup.msi
                              62 .a. -r-------- 0        0        3719     /mnt/victim_XP/Documents and
                                    Settings/All Users/Documents/desktop.ini
                            4096 .a. dr-x------ 0        0        3718     /mnt/victim_XP/Documents and
                                  Settings/All Users/Documents
Fri May 15 2009 11:57:49     6444 mac -r-------- 0        0        11028    /mnt/victim_XP/WINDOWS/Prefetch/
     WGET.EXE-0D89219B.pf
Fri May 15 2009 11:57:50     3077 mac -r-------- 0        0        11004    /mnt/victim_XP/Documents and
     Settings/test/Local Settings/Temp/U3Launcher.log
                           30720 .a. -r-------- 0        0        3145     /mnt/victim_XP/WINDOWS/system32/
                                  xcopy.exe
Fri May 15 2009 11:57:51     8192 .a. -r-------- 0        0        3514     /mnt/victim_XP/WINDOWS/system32/
     config/SAM.LOG
                          275456 .a. -r-------- 0        0        2237     /mnt/victim_XP/WINDOWS/system32/
                                  ulib.dll
                           70656 .a. -r-------- 0        0        699      /mnt/victim_XP/WINDOWS/system32/
                                  ifsutil.dll
Fri May 15 2009 11:57:52  2804224 .a. -r-------- 0        0        2233     /mnt/victim_XP/WINDOWS/system32/
     msi.dll
                          117760 .a. -r-------- 0        0        1228     /mnt/victim_XP/WINDOWS/system32/
                                  oledlg.dll
Fri May 15 2009 11:57:53    77824 .ac -r-------- 0        0        11029    /mnt/victim_XP/WINDOWS/VNCHOOKS.
     DLL
                           31232 .a. -r-------- 0        0        1371     /mnt/victim_XP/WINDOWS/system32/
                                  sc.exe
                          589824 .ac -r-------- 0        0        11030    /mnt/victim_XP/WINDOWS/WINVNC.EXE
Fri May 15 2009 11:57:54    44032 .a. -r-------- 0        0        1093     /mnt/victim_XP/WINDOWS/system32/
     msxml3r.dll
                               0 .a. dr-x------ 0        0        3629     /mnt/victim_XP/WINDOWS/system32/
                                     CatRoot2/{F750E6C3-38EE-11D1-85E5-00C04FC295EE}
                            4096 .a. dr-x------ 0        0        3617     /mnt/victim_XP/WINDOWS/system32/
                                  CatRoot2
                          351232 .a. -r-------- 0        0        3114     /mnt/victim_XP/WINDOWS/system32/
                                  winhttp.dll
                         1236480 .a. -r-------- 0        0        2777     /mnt/victim_XP/WINDOWS/system32/
                                 msxml3.dll
Fri May 15 2009 11:57:55  3153920 mac -r-------- 0        0        3632     /mnt/victim_XP/WINDOWS/system32/
     CatRoot2/{F750E6C3-38EE-11D1-85E5-00C04FC295EE}/catdb
Fri May 15 2009 11:58:03     4608 .a. -r-------- 0        0        2232     /mnt/victim_XP/WINDOWS/system32/
     msimg32.dll
                            2930 mac -r-------- 0        0        11033    /mnt/victim_XP/WINDOWS/Prefetch/
                                  LAUNCHPAD.EXE-29B57FD0.pf
                          146432 .a. -r-------- 0        0        2234     /mnt/victim_XP/WINDOWS/regedit.
                                 exe
Fri May 15 2009 11:58:04    10112 mac -r-------- 0        0        11032    /mnt/victim_XP/WINDOWS/Prefetch/
     SC.EXE-012262AF.pf
                           65024 .a. -r-------- 0        0        2350     /mnt/victim_XP/WINDOWS/system32/
                                  asycfilt.dll
                           81920 .a. -r-------- 0        0        11034    /mnt/victim_XP/Documents and
                                 Settings/test/Local Settings/Temp/~DFECA1.tmp
Fri May 15 2009 11:58:05    56832 .a. -r-------- 0        0        2235     /mnt/victim_XP/WINDOWS/system32/
     authz.dll
                           10752 .a. -r-------- 0        0        242      /mnt/victim_XP/WINDOWS/system32/
                                  clb.dll
                          114688 .a. -r-------- 0        0        2236     /mnt/victim_XP/WINDOWS/system32/
                                 aclui.dll
Fri May 15 2009 11:58:06    42496 .a. -r-------- 0        0        2257     /mnt/victim_XP/WINDOWS/system32/
     net.exe
Fri May 15 2009 11:58:07    12622 mac -r-------- 0        0        10974    /mnt/victim_XP/WINDOWS/Prefetch/
     REGEDIT.EXE-1B606482.pf
```

```
Fri May 15 2009 11:58:11   124928 .a. -r-------- 0        0        2258     /mnt/victim_XP/WINDOWS/system32/
     net1.exe
                            12288 .a. -r-------- 0        0        2188     /mnt/victim_XP/WINDOWS/system32/
                                  netrap.dll
Fri May 15 2009 11:58:13   171008 .a. -r-------- 0        0        239      /mnt/victim_XP/WINDOWS/system32/
     netmsg.dll
Fri May 15 2009 11:58:15    12664 mac -r-------- 0        0        11035    /mnt/victim_XP/WINDOWS/Prefetch/
     NET1.EXE-029B9DB4.pf
Fri May 15 2009 11:58:16    10320 mac -r-------- 0        0        11036    /mnt/victim_XP/WINDOWS/Prefetch/
     NET.EXE-01A53C2F.pf
                            28672 mac dr-x------ 0        0        28       /mnt/victim_XP/WINDOWS
Fri May 15 2009 11:58:17        0 mac dr-x------ 0        0        11038    /mnt/victim_XP/Documents and
     Settings/test/Application Data/U3/temp
                          3096576 .ac -r-------- 0        0        11039    /mnt/victim_XP/Documents and
                                Settings/test/Application Data/U3/temp/Launchpad Removal.exe
                                0 mac dr-x------ 0        0        11005    /mnt/victim_XP/Documents and
                                    Settings/test/Application Data/U3
Fri May 15 2009 11:58:18   586240 .a. -r-------- 0        0        2693     /mnt/victim_XP/WINDOWS/system32/
     mlang.dll
                             8192 mac dr-x------ 0        0        11006    /mnt/victim_XP/Documents and
                                  Settings/test/Application Data/U3/0000060327026187
Fri May 15 2009 11:58:20        0 .a. dr-x------ 0        0        10273    /mnt/victim_XP/Documents and
     Settings/test/Cookies
Fri May 15 2009 11:58:21    81920 m.c -r-------- 0        0        11034    /mnt/victim_XP/Documents and
     Settings/test/Local Settings/Temp/~DFECA1.tmp
                             9662 .ac -r-------- 0        0        11041    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/BLANK.ICO
Fri May 15 2009 11:58:22   110592 .ac -r-------- 0        0        11042    /mnt/victim_XP/WINDOWS/
     $NtUninstallKB931337$/BLAT.DLL
Fri May 15 2009 11:58:23     8192 mac -r-------- 0        0        3624     /mnt/victim_XP/WINDOWS/system32/
     CatRoot2/edb.chk
                           549376 .a. -r-------- 0        0        2971     /mnt/victim_XP/WINDOWS/system32/
                                  shdoclc.dll
                             2174 .ac -r-------- 0        0        11044    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/BLAT.LIB
                               62 .a. -r-------- 0        0        10335    /mnt/victim_XP/Documents and
                                    Settings/test/Application Data/desktop.ini
                           103936 .ac -r-------- 0        0        11043    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/BLAT.EXE
Fri May 15 2009 11:58:24     4096 .a. dr-x------ 0        0        5458     /mnt/victim_XP/Program Files/
     Internet Explorer
                          1578787 .ac -r-------- 0        0        11045    /mnt/victim_XP/WINDOWS/
                                $NtUninstallKB931337$/LIBEAY32.DLL
                           146432 .a. -r-------- 0        0        1068     /mnt/victim_XP/WINDOWS/system32/
                                  msls31.dll
                          3049472 .a. -r-------- 0        0        2306     /mnt/victim_XP/WINDOWS/system32/
                                mshtml.dll
                           632226 .ac -r-------- 0        0        11047    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/LIBSSL32.DLL
                            12212 mac -r-------- 0        0        11046    /mnt/victim_XP/WINDOWS/Prefetch/
                                  WINVNC.EXE-2B5AAA4E.pf
Fri May 15 2009 11:58:25     1490 .ac -r-------- 0        0        11051    /mnt/victim_XP/WINDOWS/
     $NtUninstallKB931337$/SBS.LNK
                            57344 .a. -r-------- 0        0        11052    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/SHORTCUT.EXE
                              113 .a. -r-------- 0        0        10364    /mnt/victim_XP/Documents and
                                    Settings/test/Local Settings/History/desktop.ini
                                0 .a. dr-x------ 0        0        10257    /mnt/victim_XP/Documents and
                                    Settings/test/Local Settings/Temporary Internet Files
                          1153024 .ac -r-------- 0        0        11048    /mnt/victim_XP/WINDOWS/
                                $NtUninstallKB931337$/OPENSSL.EXE
                           314368 .ac -r-------- 0        0        11049    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/RAR.EXE
                            65536 .ac -r-------- 0        0        11050    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/SBS.EXE
                           159232 .a. -r-------- 0        0        3182     /mnt/victim_XP/WINDOWS/system32/
                                  MSIMTF.dll
Fri May 15 2009 11:58:26    27408 mac -r-------- 0        0        11031    /mnt/victim_XP/WINDOWS/Prefetch/
     XCOPY.EXE-21FC761A.pf
                            57344 ..c -r-------- 0        0        11052    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/SHORTCUT.EXE
                            50176 .a. -r-------- 0        0        2918     /mnt/victim_XP/WINDOWS/system32/
                                  reg.exe
                            73728 .ac -r-------- 0        0        11053    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$/STUNNEL-4.11.EXE
Fri May 15 2009 11:58:27      649 mac -r-------- 0        0        11054    /mnt/victim_XP/WINDOWS/
     $NtUninstallKB931337$/send.bat
Fri May 15 2009 11:58:28     9976 mac -r-------- 0        0        10972    /mnt/victim_XP/WINDOWS/Prefetch/
     REG.EXE-0D2A95F7.pf
                             4096 ma. dr-x------ 0        0        11037    /mnt/victim_XP/WINDOWS/
                                  $NtUninstallKB931337$
                              558 mac -r-------- 0        0        11040    /mnt/victim_XP/Documents and
                                    Settings/test/Application Data/U3/0000060327026187/lplog.txt
```

```
                              93 .ac -r-------- 0        0        11055    /mnt/victim_XP/WINDOWS/
                                   $NtUninstallKB931337$/stunnel.conf
                           11264 .a. -r-------- 0        0        346      /mnt/victim_XP/WINDOWS/system32/
                                 attrib.exe
                          270214 mac -r-------- 0        0        10971    /mnt/victim_XP/System Volume
                                 Information/_restore{82B84300-9302-42B6-962A-FC10BB4357CB}/RP4/change.log
Fri May 15 2009 11:58:29     4096 ..c dr-x------ 0        0        11037    /mnt/victim_XP/WINDOWS/
     $NtUninstallKB931337$
Fri May 15 2009 11:58:31    10276 mac -r-------- 0        0        11056    /mnt/victim_XP/WINDOWS/Prefetch/
     ATTRIB.EXE-39EAFB02.pf
                          8384000 .a. -r-------- 0        0        2073     /mnt/victim_XP/WINDOWS/system32/
                                 shell32.dll
Fri May 15 2009 11:58:32  8384000 ..c -r-------- 0        0        2073     /mnt/victim_XP/WINDOWS/system32/
     shell32.dll
Fri May 15 2009 11:58:33    18944 .a. -r-------- 0        0        2242     /mnt/victim_XP/WINDOWS/system32/
     linkinfo.dll
                           143872 .a. -r-------- 0        0        2243     /mnt/victim_XP/WINDOWS/system32/
                                 ntshrui.dll
Fri May 15 2009 11:58:45     8312 mac -r-------- 0        0        11057    /mnt/victim_XP/WINDOWS/Prefetch/
     SBS.EXE-26711756.pf
Fri May 15 2009 11:58:46     1024 mac -r-------- 0        0        3503     /mnt/victim_XP/WINDOWS/system32/
     config/default.LOG
                             7158 mac -r-------- 0        0        11058    /mnt/victim_XP/WINDOWS/Prefetch/
                                  WIFIKE.EXE-34462AC6.pf
Fri May 15 2009 11:58:49     5516 mac -r-------- 0        0        11060    /mnt/victim_XP/WINDOWS/Prefetch/
     PWDUMP.EXE-242A13CC.pf
                             3686 mac -r-------- 0        0        11059    /mnt/victim_XP/WINDOWS/Prefetch/
                                  IMOKAV.EXE-3AD670C3.pf
Fri May 15 2009 11:58:54     4602 mac -r-------- 0        0        11061    /mnt/victim_XP/WINDOWS/Prefetch/
     IMOKAV.EXE-16488B1B.pf
                             4096 mac dr-x------ 0        0        10263    /mnt/victim_XP/Documents and
                                  Settings/test/Local Settings/Temp
Fri May 15 2009 11:58:55     6688 mac -r-------- 0        0        11063    /mnt/victim_XP/WINDOWS/Prefetch/
     FGDUMP.EXE-24729C4B.pf
                             5938 mac -r-------- 0        0        11062    /mnt/victim_XP/WINDOWS/Prefetch/
                                  PWDUMP.EXE-0477A877.pf
Fri May 15 2009 11:58:58     8348 mac -r-------- 0        0        11064    /mnt/victim_XP/WINDOWS/Prefetch/
     NETPASS.EXE-073AA7BD.pf
Fri May 15 2009 11:58:59    43520 .a. -r-------- 0        0        2260     /mnt/victim_XP/WINDOWS/system32/
     pstorec.dll
Fri May 15 2009 11:59:00     9200 mac -r-------- 0        0        11065    /mnt/victim_XP/WINDOWS/Prefetch/
     MAILPV.EXE-13087CD0.pf
Fri May 15 2009 11:59:02    32768 .a. -r-------- 0        0        10325    /mnt/victim_XP/Documents and
     Settings/test/Local Settings/History/History.IE5/index.dat
Fri May 15 2009 11:59:03    11256 mac -r-------- 0        0        11066    /mnt/victim_XP/WINDOWS/Prefetch/
     IEPV.EXE-0CC66364.pf
Fri May 15 2009 11:59:05        0 .a. dr-x------ 0        0        10340    /mnt/victim_XP/Documents and
     Settings/test/Local Settings/Application Data/Microsoft/Credentials
                             8110 mac -r-------- 0        0        11067    /mnt/victim_XP/WINDOWS/Prefetch/
                                  MSPASS.EXE-00EB4D42.pf
                                0 .a. dr-x------ 0        0        10341    /mnt/victim_XP/Documents and
                                     Settings/test/Local Settings/Application Data/Microsoft/Credentials/S
                                     -1-5-21-515967899-706699826-725345543-1003
                                0 .a. dr-x------ 0        0        10343    /mnt/victim_XP/Documents and
                                     Settings/test/Application Data/Microsoft/Credentials/S
                                     -1-5-21-515967899-706699826-725345543-1003
                                0 .a. dr-x------ 0        0        10342    /mnt/victim_XP/Documents and
                                     Settings/test/Application Data/Microsoft/Credentials
Fri May 15 2009 11:59:07    33280 .a. -r-------- 0        0        2126     /mnt/victim_XP/WINDOWS/system32/
     cryptdll.dll
                           415744 .a. -r-------- 0        0        2128     /mnt/victim_XP/WINDOWS/system32/
                                  samsrv.dll
                           721920 .a. -r-------- 0        0        2125     /mnt/victim_XP/WINDOWS/system32/
                                  lsasrv.dll
Fri May 15 2009 11:59:08    13379 mac -r-------- 0        0        10223    /mnt/victim_XP/WINDOWS/system32/
     wbem/Logs/wbemess.log
                             7570 mac -r-------- 0        0        11068    /mnt/victim_XP/WINDOWS/Prefetch/
                                  CACHEDUMP.EXE-2003DDD8.pf
Fri May 15 2009 11:59:11     8890 mac -r-------- 0        0        11069    /mnt/victim_XP/WINDOWS/Prefetch/
     PSPV.EXE-33538890.pf
Fri May 15 2009 11:59:13    98304 .a. -r-------- 0        0        2414     /mnt/victim_XP/WINDOWS/system32/
     cscript.exe
                             7634 mac -r-------- 0        0        11070    /mnt/victim_XP/WINDOWS/Prefetch/
                                  PRODUKEY.EXE-0AF6A5C7.pf
Fri May 15 2009 11:59:18    32768 mac -r-------- 0        0        11072    /mnt/victim_XP/Documents and
     Settings/test/Local Settings/History/History.IE5/MSHist012009050420090511/index.dat
                                0 ..c dr-x------ 0        0        10273    /mnt/victim_XP/Documents and
                                     Settings/test/Cookies
                            16384 m.. -r-------- 0        0        10859    /mnt/victim_XP/Documents and
                                  Settings/test/Local Settings/History/History.IE5/MSHist012009051520090516/
                                  index.dat
                                0 .ac dr-x------ 0        0        10264    /mnt/victim_XP/Documents and
                                     Settings/test/Local Settings/History
```

```
                            16384 m.c -r-------- 0        0        10332    /mnt/victim_XP/Documents and
                                  Settings/test/Cookies/index.dat
                                0 mac dr-x------ 0        0        11071    /mnt/victim_XP/Documents and
                                      Settings/test/Local Settings/History/History.IE5/
                                      MSHist012009050420090511
                           786432 m.c -r-------- 0        0        10132    /mnt/victim_XP/Documents and
                                  Settings/test/NTUSER.DAT
                            32768 m.c -r-------- 0        0        10320    /mnt/victim_XP/Documents and
                                  Settings/test/Local Settings/Temporary Internet Files/Content.IE5/index.dat
                                0 ..c dr-x------ 0        0        10257    /mnt/victim_XP/Documents and
                                      Settings/test/Local Settings/Temporary Internet Files
                            32768 m.c -r-------- 0        0        10325    /mnt/victim_XP/Documents and
                                  Settings/test/Local Settings/History/History.IE5/index.dat
                                0 .ac dr-x------ 0        0        10258    /mnt/victim_XP/Documents and
                                      Settings/test/Local Settings/Temporary Internet Files/Content.IE5
Fri May 15 2009 11:59:19        0 mac dr-x------ 0        0        10858    /mnt/victim_XP/Documents and
     Settings/test/Local Settings/History/History.IE5/MSHist012009051520090516
                            16384 .ac -r-------- 0        0        10859    /mnt/victim_XP/Documents and
                                  Settings/test/Local Settings/History/History.IE5/MSHist012009051520090516/
                                  index.dat
                             4096 mac dr-x------ 0        0        10265    /mnt/victim_XP/Documents and
                                   Settings/test/Local Settings/History/History.IE5
                            10414 mac -r-------- 0        0        11073    /mnt/victim_XP/WINDOWS/Prefetch/
                                  CSCRIPT.EXE-1C26180C.pf
Fri May 15 2009 11:59:21     7714 mac -r-------- 0        0        11074    /mnt/victim_XP/WINDOWS/Prefetch/
     WUL.EXE-0C129EB7.pf
Fri May 15 2009 11:59:26     1024 mac -r-------- 0        0        3501     /mnt/victim_XP/WINDOWS/system32/
     config/system.LOG
Fri May 15 2009 11:59:53    26112 .a. -r-------- 0        0        3095     /mnt/victim_XP/WINDOWS/system32/
     vdmdbg.dll
Fri May 15 2009 11:59:54    25600 .a. -r-------- 0        0        1529     /mnt/victim_XP/WINDOWS/system32/
     utildll.dll
                           135680 .a. -r-------- 0        0        3042     /mnt/victim_XP/WINDOWS/system32/
                                  taskmgr.exe
Fri May 15 2009 11:59:55   135680 ..c -r-------- 0        0        3042     /mnt/victim_XP/WINDOWS/system32/
     taskmgr.exe
Fri May 15 2009 12:00:00     1024 mac -r-------- 0        0        10150    /mnt/victim_XP/Documents and
     Settings/test/NTUSER.DAT.LOG
Fri May 15 2009 12:00:03    36864 mac dr-x------ 0        0        10207    /mnt/victim_XP/WINDOWS/Prefetch
                            16126 mac -r-------- 0        0        11075    /mnt/victim_XP/WINDOWS/Prefetch/
                                  TASKMGR.EXE-20256C55.pf
Fri May 15 2009 12:00:12      276 .a. -r-------- 0        0        10934    /mnt/victim_XP/agent/stop.bat
                             8192 m.c -r-------- 0        0        3514     /mnt/victim_XP/WINDOWS/system32/
                                   config/SAM.LOG
Fri May 15 2009 12:00:13   388608 .ac -r-------- 0        0        2241     /mnt/victim_XP/WINDOWS/system32/
     cmd.exe
Fri May 15 2009 12:00:23    12060 mac -r-------- 0        0        10449    /mnt/victim_XP/WINDOWS/Prefetch/
     CMD.EXE-087B4001.pf
Fri May 15 2009 12:00:26     4096 mac dr-x------ 0        0        10846    /mnt/victim_XP/agent
                          83717408 m.c -r-------- 0        0        11003    /mnt/victim_XP/agent/log.pml
Fri May 15 2009 12:00:32     1024 mac -r-------- 0        0        3502     /mnt/victim_XP/WINDOWS/system32/
     config/software.LOG
Fri May 15 2009 12:00:33   102140 mac -r-------- 0        0        10927    /mnt/victim_XP/WINDOWS/Prefetch/
     PROCMON.EXE-09DDAFE0.pf
Fri May 15 2009 12:00:41 31188785 mac -r-------- 0        0        11076    /mnt/victim_XP/agent/log.csv
```

Policy for bruk av

Kvalitetssikring av Exploits

Formål

Formålet med dette dokumentet er å gi retningslinjer for bruk av rammeverket KSE ved kvalitetssikring av exploits.

Omfang

Denne sikkerhetspolicyen gjelder alle faste og midlertidig ansatte, innleide konsulenter og andre som benytter KSE på vegne av <firma>

Bruk av KSE

Ansatte i <firma> kan benytte seg av KSE ved testing og kvalitetssikring av exploits. Ved å bruke KSE forplikter alle ansatte seg til å følge regler angitt i dette dokumentet.

Sikring av testmiljø

Alle ansatte plikter å sikre testmiljøet tilstrekkelig ved bruk av KSE. Maskinen hvor testen skal utføres skal være fysisk koblet fra nettverket. Det kan gis unntak hvis testen utføres i et virtuel miljø eller hvis man ved hjelp av brannmur kan restriktere nettverkstrafikken.

Kvalitetssikring av test

Alle tester må utføres to eller flere ganger for å verifisere resultatene. Eventuelle avik mellom testene skal komme frem i exploitens dokumentasjon.

Dokumentasjon

Det skal utarbeides en rapport etter utført test som minimum har med følgende data:

- Testens navn
- Testens dato
- Navn på ansatt som utførte testen
- Beskriver av exploitens antatte virkemåte
- Alle logger og rapporter generert av KSE
- Eventuelle avik ved gjentatte tester

Ved brudd av policy

Ved brudd på denne policyen vil ansatte risikere reprimande og i verst fall oppsigelse.

Definisjoner

KSE	Kvalitetssikring av Exploits

Brukerveiledning for

Kvalitetssikring av Exploits

Innhold

Om KSE

KSE (Kvalitetssikring av Exploits) er et rammeverk for å monitorere en exploit for å se hvordan denne oppfører seg. Ved å se hvilke endringer exploiten påfører systemet ressurser, kan man finne ut om exploiten gjør det den skal, eller om den inneholder ondsinnet kode.

Systemkrav

For å kjøre KSE anbefales følgende minimum systemkrav:

Operativsystem	Debian Linux Lenny eller nyere
Prosessor	Pentium 4 1.6 GHz eller nyere
Minne	2 GB eller mer
Harddisk	20 GB ledig plass

Følgende software må også installeres:

Debian-pakker:
- php5-cli
- apache2
- libapache2-mod-php5
- php5-sqlite
- sqlite
- build-essensials
- libfuse2
- fuse-utils

Annet:
- Vmware Server 2
- Windows XP Servicepack 2

Installasjon

Operativsystem

1. Installer Debian Linux fra debian.com. Velg vekk alle GUI-komponenter under installasjonen.
2. Installer alle påkrevde pakker ved hjelp av pakkesystemet. Eks: apt-get install <pakkenavn>
3. Kopier inn KSE-filene til /usr/local/kse

Vmware

1. Last ned og installer Vmware Server 2 fra vmware.com
2. Kjør vmware-config.pl og velg standardinnstillinger på alt. Dette vil blant annet gjøre at vmnet1 blir satt som et host-only nettverk, noe som er nødvendig for at maskinene ikke skal kunne gå på internett, men likevel skal kunne kommunisere med hverandre og hosten.
3. Legg inn “modprobe fuse” nederst i /etc/modules. Dette er nødvendig for å kunne montere virtuelle harddisker.
4. Opprett to virtuelle maskiner. En som har rollen som angriper og en som har rollen som offer. Installert Windows XP på begge maskinene. Sørg for at angriperen får navnet “Attacker_WinXP” og offeret får navnet “Victim_WinXP”.

Webside

Opprett /etc/apache2/conf.d/kse.conf med følgende tekst:

```
Alias /kse /usr/local/kse/shared

<Directory /usr/local/kse/shared>
        Options Indexes FollowSymLinks
        DirectoryIndex index.php
</Directory>
```

Utføre test

Klargjøre for test

Før testen kan utføres må testmiljøet gjøres klart. Løsningen kommer med programmer som utfører de nødvendige oppstartrutinene. Kjør følgene kommando for å starte:

```
# /usr/local/kse/bin/autoscript.sh <navn på test>
```

Denne kommandoen vil utføre følgende operasjoner:

- Ta et “snapshot” av den virtuelle maskinen
- Montere den virtuelle harddisken og lage en database over alle filene (MD5, MAC)
- Starte operativsystemet
- Starte agenten som samler inn data

Når alle oppstartsrutinene er feridig, vil man få beskjed om at man skal utføre testen, og trykke en tast for å fortsette når man er ferdig.

Utføre test

Det finnes to måter å utføre selve testen:

1. Logg inn på den virtuelle maskinen for å ta kontroll over konsollsesjonen. Dette gjøres ved å koble til Vmwares webgrensesnittet. Åpne internettleseren på http://<serverip>:8333. Logg inn med brukernavnet og passordet satt ved installasjonen av Vmware. Velg den virtuelle maskinen som heter «Victim_WinXP» og trykk på «Console».
2. Vmware Server har et verktøy for å utføre visse operasjoner på den virtuelle maskinen fra utsiden, dvs. serveren Vmware kjører på. For å overføre en fil til den virtuelle maskinen vil man typisk bruke en kommando som:

   ```
   vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp "test"
   copyFileFromHostToGuest "Victim_WinXP.vmx" /tmp/exploit.exe "C:\temp\"
   ```

   For å starte en fil på den virtuelle maskinen må man bruke følgende kommando:

   ```
   vmrun -h https://127.0.0.1:8333/sdk -u root -p testbox -gu test -gp "test" runProgramInGuest
   "Victim_WinXP.vmx" "C:\temp\exploit.exe"
   ```

Når man er ferdig med å kjøre exploiten, må man trykke en tast for å fortsette med analysen. Dette vil utføre resten av testen og generere logger til brukeren.

Resultat

Loggfiler

Under analysen blir det generelt en rekke loggfiler av filaktivitet, nettverksaktivitet, etc. Disse loggene blir lagret i /usr/local/kse/logs.

Rapport

Det er også mulig å se resultatene av testen i et eget webgrensesnitt. For å aksessere dette, gå på http://<serverip>/kse

HØGSKOLEN I GJØVIK

PROSJEKTAVTALE

mellom Høgskolen i Gjøvik (HiG) (utdanningsinstitusjon), Ernst & Young AS, og
Truls Hagen, Gudmund Nordstrøm, Mathias Bjerke og Bjørnar Prestaasen (student(er)).

Avtalen angir avtalepartenes plikter vedrørende gjennomføring av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer:

1. Studenten(e) skal gjennomføre prosjektet i perioden fra 12/1-09 til 3/6 -09.

   Studentene skal i denne perioden følge en oppsatt fremdriftsplan der HiG yter veiledning.
   Ernst & Young AS yter avtalt prosjektbistand til fastsatte tider. Ernst & Young AS stiller til rådighet kunnskap og annet intellektuelt materiale hentet fra Ernst & Youngs metodeverk som bidrag til gjennomføring av prosjektet. Det forutsettes at de gitte problemstillinger det arbeides med er aktuelle og på et nivå tilpasset studentenes faglige kunnskaper. Ernst & Young AS plikter på forespørsel fra HiG å gi en vurdering av prosjektet vederlagsfritt.

2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:
   - Ernst & Young AS dekker sine egne kostnader i forbindelse med gjennomføringen av prosjektet. Studentene dekker utgifter for trykking og ferdigstillelse av den skriftlige besvarelsen vedrørende prosjektet.
3. Eiendomsretten til eventuell prototyp tilfaller den som har betalt komponenter og materiell mv. som er brukt til prototypen. Dersom det er nødvendig med større og/eller spesielle investeringer for å få gjennomført prosjektet, må det gjøres en egen avtale mellom partene om eventuell kostnadsfordeling og eiendomsrett.

4. HiG står ikke som garantist for at det Ernst & Young AS har bestilt fungerer etter hensikten, ei heller at prosjektet
   blir fullført. Prosjektet må anses som en eksamensrelatert oppgave som blir bedømt av faglærer/veileder og sensor. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.

5. Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode, disketter, taper mv. som inngår som del av eller vedlegg til besvarelsen, gis det en kopi av til HiG, som vederlagsfritt kan benyttes til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av HiG til andre formål, og ikke overlates til utenforstående uten etter avtale med de øvrige parter i denne avtalen. Dette gjelder også firmaer hvor ansatte ved HiG og/eller studenter har interesser.

   Besvarelser med karakter C eller bedre registreres og plasseres i skolens bibliotek. Det legges også ut en elektronisk prosjektbesvarelse uten vedlegg på bibliotekets del av skolens Internett-sider. Dette avhenger av at studentene skriver under på en egen avtale hvor de gir biblioteket tillatelse til at deres hovedprosjekt blir gjort tilgjengelig i papir og nettutgave (jfr. Lov om opphavsrett). Ernst & Young AS og veileder godtar slik offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og dekan om de i løpet av prosjektet endrer syn på slik offentliggjøring.

6. Besvarelsens spesifikasjoner og resultat kan anvendes i Ernst & Young ASs egen virksomhet. Gjør studenten(e) i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom Ernst & Young AS og student(er) bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.

BP
GN
TH
MB

7. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten(e) ikke rett til å publisere sin besvarelse, det være seg helt eller delvis eller som del i annet arbeide, uten samtykke fra Ernst & Young AS. Tilsvarende samtykke må foreligge i forholdet mellom student(er) og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.

8. Studenten(e) leverer 3 - tre - eksemplarer av oppgavebesvarelsen med vedlegg til Studenttorget. I tillegg leveres et eksemplar til Ernst & Young AS. HiG kan stille til disposisjon ytterligere eksemplar(er) for Ernst & Young AS mot at denne godtgjør produksjonskostnadene.

9. Denne avtalen utferdiges med et eksemplar til hver av partene. På vegne av HiG er det dekan som godkjenner avtalen.

10. I det enkelte tilfelle kan det inngås egen avtale mellom Ernst & Young AS, student(er) og HiG som nærmere regulerer forhold vedrørende bl.a. eiendomsrett, videre bruk, konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene.

    Dersom Ernst & Young AS og student(er) ønsker en videre eller ny avtale, skjer dette uten HiG som partner.

11. Når HiG også opptrer som oppdragsgiver trer HiG inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.

12. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene i mellom. Dersom det ikke oppnås enighet, er partene enige om at tvisten løses av voldgift, etter bestemmelsene i tvistemålsloven av 13.8.1915 nr. 6, kapittel 32.

13. Deltakende personer ved prosjektgjennomføringen:

HiGs veileder (navn): ANDRÉ ÅRNES

Ernst & Young AS kontaktperson (navn): Eirik R. Thormodsrud

Student(er) (signatur): Bjørnar Prestaasen dato 11/3-09

Gudmund Nordstrøm dato 11/3-09

Truls Haugen dato 11/3-09

Mathias Bjelke dato 11/3-09

Ernst & Young AS (signatur): dato 10/3-09

Dekan (signatur): Kai Robert Jakobsen dato 11/3-09

Revidert 11.10.07, Ivar Moe

Bjørnar Prestaasen, Gudmund Nordstrøm,
Mathias Bjerke og Truls Hagen

FORPROSJEKTRAPPORT

Kvalitetssikring av Exploits

Innhold

1 Mål og rammer

De siste årene har vi hatt en enorm utvikling innen informasjonteknologi. De fleste bedrifter har tatt i bruk IT på ett eller flere områder. Dessverre har ikke utviklingen av informasjonssikkerhet holdt samme tempo, noe som har gitt store, usikrede IT-installasjoner.

1.1 Innledning

Vår oppdragsgiver, Ernst & Young, utfører IT-revisjon og penetrasjonstesting av systemene til sine kunder. Det finnes en rekke verktøy og teknikker for å utføre denne testingen, men dessverre kan ikke alle brukes av frykt for å skade kundens datasystemer. Exploits er en av disse.

En exploit er en programkode, en spesielt formatert tekststreng eller annen data spesielt laget for å utnytte et sikkerhetshull i en programvare. Disse er ofte fritt tilgjengelige på Internett og kan testes og benyttes av alle.

EY har et ønske om å ta i bruk exploits under penetrasjonstesting hos sine kunder. Pr. i dag er den eneste måten å kvalitetssikre exploits å gå gjennom kildekoden linje for linje, noe som er svært tidkrevende.

Vår løsning vil kunne gi EY muligheten til å teste nye exploits i et sikkert miljø, slik at de får en oversikt over hva exploiten gjør. Dette medfører at EY selv kan teste og godkjenne nye exploits til egen bruk.

1.2 Tidligere arbeid

Gjennom en mindre undersøkelse kan det tyde på at det er få åpne prosjekter som omhandler analyse og kvalitetssikring av exploits. Det er derimot skrevet om analyse av malware [1] og testing av antivirusverktøy [2]. Det finnes også nettsider som tilbyr online analysering av malware [3].

1.3 Effektmål

Prosjektets effektmål er å gi Ernst & Young et hjelpemiddel de kan benytte for å teste og kvalitetssikre exploits, slik at disse kan brukes ved penetrasjonstesting ute hos kunder. Dette vil gi EY mulighet for å utføre en mer omfattende test og øke kvaliteten på sitt produkt.

[1] Malware analysis - Petter Langeland Wedum, NTNU
[2] Automated testing of antivirus coverage - Lars Haukli, NTNU
[3] Online analyse av malware - CWSandbox (http://www.cwsandbox.org)

1.4 Resultatmål

- Det skal utarbeides et konsept/rammeverk for testing av exploits i et virtuelt testmiljø. Rammeverket skal være så generelt at det kan implementeres på forskjellige operativsystemer.
- Konseptet skal i eksperimentet implementeres på Windows XP, og testes mot en fiktiv, en kjent og en ny exploit.
- Prosjektrapporten vil være dokumentasjonen for rammeverket/konseptet, samt implementasjonen/eksperimentet.
- Det skal lages en mal for rapporten som genereres etter analyse.
- Brukerveiledning og policy må lages for at brukerne skal kunne benytte løsningen på en trygg og effektiv måte.

1.5 Læringsmål

- Gruppen har som mål å få dypere forståelse rundt analysering av skadelig programvare gjennom både teori og praksis.
- Gruppen vil få erfaring med verktøy for testing i et virtuelt miljø og verktøy for analysering ressursene angitt i oppgavebeskrivelsen.
- Gjennom praktisk bruk av verktøy for prosjektstyring og revisjonskontroll vil gruppemedlemmene få erfaring i å drive prosjekter, en arbeidsform som er veldig utbredt i dag.

1.6 Rammer

- Prosjektrapporten skal leveres til kopisentralen den 20.05.2009 og fremføres 03.06.2009.
- Ernst & Youngs ønsker for funksjonalitet og dokumentasjon må oppfylles.
- Skolens formelle krav for hovedprosjektet må oppfylles.

2 Omfang

Dette kapittelet gir en kort beskrivelse av oppgavens innhold og avgrensninger. Under oppgavebeskrivelse finnes en prioritert liste over hvilke ressurser som skal overvåkes.

2.1 Oppgavebeskrivelse

Ved penetrasjonstesting er det ofte nødvendig/ønskelig å benytte seg av exploits eller annen potensielt skadelig programvare for å gjøre en mer grundig test. Koden som skal kjøres må ha en tilstrekkelig grad av sikkerhet, slik at man kan vite at koden kun gjør det den skal. Det er derfor ønskelig å kunne teste koden i et sikkert miljø hvor man kan overvåke endringer i målsystemet for å verifisere kodens funksjon, uten nødvendigvis å måtte lese gjennom hele kildekoden. Prosjektgruppen skal på grunnlag av dette utvikle en løsning som overvåker potensielt skadelig programvare i et kontrollert testmiljø. Dette testmiljøet vil bli kontrollert og overvåket på to måter: i sanntid og ved analyse før og etter utført test. Resultatet skal til slutt brukes for å finne ut hvordan gitte ressurser hos målsystemet er blitt påvirket av koden.

Kode som testes skal kunne bli eksekvert enten lokalt eller fra en annen maskin via nettverket. Når det gjelder målsystemet skal det fokuseres på følgende ressurser, i prioritert rekkefølge:

1. Endringer i filer
2. Prosesser (startede og avsluttede)
3. Nettverk (tilstand og trafikk)
4. Endringer i Windows register
5. Endringer i internminnet

Målet er få en verifikasjon på at exploiten gjør det den skal, og ikke påvirker andre prosesser som for eksempel å innføre rootkits eller andre uønskede programmer. Det skal videre utarbeides guider og en policy for hvordan tester skal gjennomføres og rapporteres. Dokumentasjonen bør være generell nok til å kunne benyttes på flere systemer, og den bør dekke hvordan systemet skal settes opp, benyttes og tilbakestilles for å kunne ha et rent testmiljø til en hver tid.

Oppgaven vil i hovedsak gå ut på å utvikle et generelt rammeverk med tanke på muligheter for videreføring til andre plattformer (Linux, OS X, med mer), så prosjektgruppen bør av den grunn vise produktuavhengighet i så stor grad som mulig.

2.2 Avgrensninger

- I det virtuelle testmiljøet har vi valgt å kun benytte Windows XP som målsystem. Windows er pr. i dag det mest brukte operativsystemet [4], noe som sannsynligvis gjør det et mer attraktivt mål for angripere som i sin tur igjen genererer fler exploits.
- Testmiljøet vil kun bestå av én angriper og ett offer.
- Kun testing innenfor et virtuelt miljø.
- Vi vil kun fokusere på testing av exploits da det finnes mange lignende prosjekter for testing av annen malware.

[4] www.w3counter.com/globalstats.php

3 Prosjektorganisering

Dette kapittelet viser hvilke personer som er involvert i prosjektet og hvilke ansvar og roller disse har.

3.1 Ansvarsforhold

Truls Hagen:	Prosjektleder. Ansvar for fremdrift og fordeling av oppgaver.
Gudmund Nordstrøm:	Dokumentansvarlig. Ansvar for fremdrift og ferdigstilling av dokument.
Bjørnar Prestaasen:	Kontaktperson. Ansvar for å distribuere informasjon.
Mathias Bjerke:	Informasjonsansvarlig. Ansvar for web-side og prosjektstyringsverktøy.

3.2 Øvrige roller og bemanning

Oppdragsgiver:	Ernst & Young
Kontaktperson:	Eirik Thormodsrud
Veileder:	André Årnes

3.3 Grupperegler

- Arbeid med hovedprosjektet foregår normalt fra 08:00 til 15:00, med unntak av mandager hvor det kun arbeides frem til lunsj. Dette resulterer i ca. 30 timer pr. uke. Avvik fra dette avtales innad i gruppen.
- Studentene plikter å følge avtalt arbeidstid og håndtere oppgaver som blir tildelt utover i prosjektet.
- Arbeid vil vanligvis foregå i A032. Ved arbeid hjemmefra/andre steder avtales dette på forhånd.
- Alle uenigheter og problemer rundt oppgaven skal diskuteres og løses innad i gruppen. Hvis uenighet ikke løses ved avstemning har prosjektleder dobbeltstemme.
- Det skal daglig føres logg over hva som er blitt gjort samt antall timer brukt.
- Ved fravær grunnet sykdom eller annet skal det meldes fra til prosjektleder samme dag.

4 Planlegging, oppfølging og rapportering

Dette kapittelet tar for seg inndeling og arbeidsmåter i prosjektet.

4.1 Hovedinndeling av prosjektet

Vi skal lage et konseptuelt rammeverk for kvalitetssikring av exploits som skal implementeres/testes ut i et testmiljø. Når vi skal utvikle testmiljøet vil vi logge mest mulig relevante data, men prøve å få minst mulig falske positiver. Vi har også prioritert rekkefølgen på ressursene som skal overvåkes.

4.2 Valg av systemutviklingsmodell

Når vi med bakgrunn i hovedinndelingen ser på hvilke systemutviklingsmodeller som passer prosjektet, er det naturlig å se på de sekvensielle modellene, siden vi skal jobbe oss fremover ved å lage del for del. En iterativ modell som for eksempel eXtreme Programming eller en evolusjonær modell vil gi tidlige delleveranser, men det er viktig for oss at hver delleveranse blir ferdigstilt i prioritert rekkefølge. Ved å lage ferdig én og én del, spesielt når det kommer til de forskjellige ressursene som skal overvåkes, vil vi kunne gjenbruke mye fra tidligere delleveranser. Siden hver del i seg selv ikke skal leveres, men kun testes og jobbes videre med, vil vi heller ikke dra nytte av del-leveringene til de smidige utviklingsmodellene. For å være sikre på å få testet ut de delene av rammeverket som vi forsøker å implementere, er det viktig at vi ser på de sekvensielle utviklingsmodellene. Det er ganske klart hva hver del skal gjøre, så vi vil ikke dra nytte av den kreative delen til de iterative modellene.

Den sekvensielle modellen fossefall gir god styring og er lett å bruke, men fossefall er for risikofylt ved at hele prosjektet kommer i mål samtidig. Det kan med andre ord bli vanskelig å prioritere bort deler av målene før vi er ferdig med hele prosjektet, hvis tiden ikke skulle strekke til.

Vi velger derfor å jobbe etter en inkrementell modell siden den gjør at vi først kan jobbe med det konseptuelle, for så å ta for oss testmiljøet og ressursene som skal overvåkes. En inkrementell modell vil være en bedre arbeidsmodell for dette prosjektet, da den jobber seg sekvensielt fremover.

4.3 Plan for statusmøter og beslutningspunkter

Det skal holdes interne statusmøter hver fredag. Med jevne mellomrom holdes statusmøter sammen med veileder Andre Aarnes. Dette kan gjøres over telefon. To større møter holdes sammen med veileder og oppdragsgiver. Disse holde på Gjøvik og er henholdsvis 20. februar og 17. april.

Beslutninger tas ved avstemning og eventuelt i samråd med veileder. Ved likt antall stemmer har prosjektleder dobbeltstemme.

5 Organisering av kvalitetssikring

Dette kapittelet beskrive hvilke risikoer som finnes og hvilke tiltak man må innføre for å unngå eventuelle problemer.

5.1 Dokumentasjon, standardbruk og kildekode

Vi bruker LaTex som publiseringsspråk for selve rapporten. Vi unngår dermed mye formateringsjobb samtidig som vi kan bruke konfigurasjonsstyring siden rapporten består av rene tekstfiler.

5.2 Konfigurasjonsstyring

Som konfigurasjonsverktøy bruker vi Subversion. Ved bruk av Subversion kan flere brukere jobbe med rapporten samtidig uten å skape konflikter. Dette gjøres for å ha bedre oversikt over de forskjellige dokumentversjonene.

Vi bruker Trac som er et prosjektstyringsverktøy sammen med Subversion for å holde orden på de forskjellige versjonene. I Trac har vi loggbok, tidslinje over sist endrede dokumenter og en oversikt over hvilke oppgaver som gjenstår. Ved hjelp av disse verktøyene har vi til enhver tid full oversikt over hvilke oppgaver som er utført og hva som gjenstår.

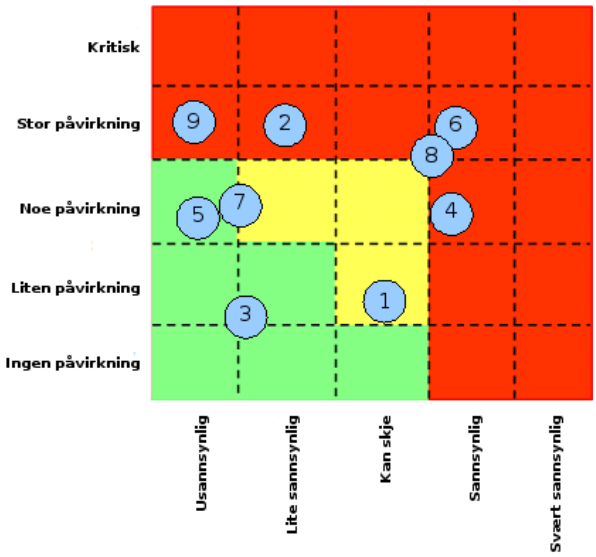
5.3 Risikoanalyse

Identifisering, analyse og enkle tiltak til risikoer:

Tabell 1: Risikoanalyse

ID	Hendelse	Sannsynlig	Konsekvens	Tiltak
1	Ødelagt testmiljø	Kan skje	Liten påvirkning	Regelmessig sikkerhetskopiering
2	Sykdom	Lite sannsynlig	Stor påvirkning	Subversion, Trac, nøkkelpersoner
3	Uenighet/krangel	Lite sannsynlig	Liten påvirkning	Grupperegler
4	Holder ikke tidsfrister	Sannsynlig	Noe påvirkning	Gantt og statusmøter
5	Mangel på oppfølging	Usannsynlig	Noe påvirkning	Opprettholde dialog
6	Mangel på intern kompetanse	Sannsynlig	Stor påvirkning	Bruke EY, veileder samt relevant litteratur
7	Krav til endring	Lite sannsynlig	Noe påvirkning	Klare avgrensninger og rammer
8	For dårlig utstyr	Sannsynlig	Stor påvirkning	Oppgradering av noe utstyr
9	Miste rapport	Usannsynlig	Kritisk	Sikkerhetskopi rutiner og Subversion

Figur 1: Risikomatrise



Tiltak og oppfølging av de mest kritiske risikoene:

- 2 - Sykdom over lengre perioder. For å forhindre at sykdom på gruppemedlemmer skal påvirke prosjektet, har vi innført noen tiltak. Ved hjelp av Trac har vi god oversikt over hvilke oppgaver gruppemedlemmene har til enhver tid. Det er også viktig å unngå nøkkelpersoner for å hindre at vi mister viktig kompetanse hvis en av gruppemedlemmene faller fra. Dette gjøres ved kontinuerlig dokumentasjon og at alle har innsikt i arbeidet.
- 4 - Holder ikke tidsfrister. Ved å ha realistiske mål og gode avgrensninger kan tidsproblemer reduseres betraktelig. Vi har også tatt høyde for at de siste inkrementene i testperioden, som for eksempel minneendringer, kan utgå. Ukentlige statusmøter hjelper oss å fokusere på fremdriften i selve prosjektet.
- 6 - Teknisk vanskelig/kompetanse. For å minske denne risikoen vil vi benytte oss av den kompetansen EY og veileder innehar på området. I tillegg vil vi søke informasjon i litteratur og på Internett.
- 8 - For dårlig utstyr. Vi har valgt å teste utstyret vi har fått tildelt tidlig, for å finne ut om det er behov for å få tak i bedre utstyr. Dette medførte at vi måtte låne en bedre test-PC enn det vi fikk av skolen og kjøpe RAM til denne.
- 9 - Miste rapport. For å minske denne risikoen for å miste rapporten bruker vi Subversion. Alle gruppemedlemmene har en kopi på sin pc av alt som er skrevet på selve rapporten. I tillegg ligger alt på en sentral server som det blir tatt sikkerhetskopi av hver natt.

6 Plan for gjennomføring

Vedlagt ligger Gantt-skjema som viser oversikt og planlagt gjennomføring. Skjemaet viser i hovedsak følgende punkter:

- Liste over aktiviteter
- Milepæler
- Beslutningspunkter
- Tids- og ressursplaner

6.1 Gantt-skjema

ID	Task Name	Duration	Start	Finish	Predecessors
1	**Forprosjekt**	**13 days**	**Wed 14.01.09**	**Fri 30.01.09**	
2	Lese lignende oppgaver	10 days	Wed 14.01.09	Tue 27.01.09	
3	Lage webside	5 days	Mon 19.01.09	Fri 23.01.09	
4	Etablere KSE testmiljø	5 days	Mon 19.01.09	Fri 23.01.09	
5	Ferdig oppsatt testmiljø	0 days	Fri 23.01.09	Fri 23.01.09	4
6	Førsteutkast forprosjektrapport	0 days	Fri 23.01.09	Fri 23.01.09	
7	Møte veileder	0 days	Fri 23.01.09	Fri 23.01.09	
8	Lage outline til hovedrapport	5 days	Mon 26.01.09	Fri 30.01.09	
9	Redigering/Gjennomlesing av forpros	5 days	Mon 26.01.09	Fri 30.01.09	
10	**Levere forprosjekt**	0 days	Fri 30.01.09	Fri 30.01.09	
11	Levere prosjektavtale	0 days	Fri 30.01.09	Fri 30.01.09	
12	**Hoveddel**	**86 days**	**Mon 26.01.09**	**Mon 25.05.09**	
13	Konseptuell fase	16 days	Mon 26.01.09	Fri 13.02.09	
14	Møte veileder	0 days	Fri 06.02.09	Fri 06.02.09	
15	Filer	10 days	Mon 09.02.09	Fri 20.02.09	
16	Utkast rapport	0 days	Wed 18.02.09	Wed 18.02.09	
17	**Møte veileder og E&Y**	0 days	Fri 20.02.09	Fri 20.02.09	
18	Prosesser	10 days	Mon 23.02.09	Fri 06.03.09	
19	Møte veileder	0 days	Fri 13.03.09	Fri 13.03.09	
20	Nettverk	10 days	Mon 09.03.09	Fri 20.03.09	
21	Register	10 days	Mon 23.03.09	Fri 03.04.09	
22	Minne	10 days	Mon 06.04.09	Fri 17.04.09	
23	Møte veileder	0 days	Fri 03.04.09	Fri 03.04.09	
24	Utkast rapport	0 days	Mon 13.04.09	Mon 13.04.09	
25	**Møte veileder og E&Y**	0 days	Fri 17.04.09	Fri 17.04.09	24
26	Autogenerert rapport	10 days	Mon 20.04.09	Fri 01.05.09	
27	Policy og brukerveiledning	10 days	Mon 20.04.09	Fri 01.05.09	
28	Utkast rapport veileder og E&Y	0 days	Wed 06.05.09	Wed 06.05.09	
29	Møte veileder	0 days	Fri 08.05.09	Fri 08.05.09	
30	Ferdigstille rapport	8 days	Mon 11.05.09	Wed 20.05.09	
31	**Levere rapport kopisentral**	**0 days**	**Wed 20.05.09**	**Wed 20.05.09**	30
32	Levere rapport kopisentral	0 days	Mon 25.05.09	Mon 25.05.09	
33	**Plakat**	**8 days**	**Thu 21.05.09**	**Tue 02.06.09**	
34	Lage plakat	6 days	Thu 21.05.09	Thu 28.05.09	
35	Levere kopisentral	0 days	Thu 28.05.09	Thu 28.05.09	34
36	Levere studenttorget	0 days	Tue 02.06.09	Tue 02.06.09	
37	**Presentasjon**	**5 days**	**Wed 27.05.09**	**Wed 03.06.09**	
38	Forberede presentasjon	4 days	Wed 27.05.09	Mon 01.06.09	
39	Presentasjon	0 days	Wed 03.06.09	Wed 03.06.09	38

ID	Task Name	Duration	Start	Finish	Predecessors
1	**Forprosjekt**	**13 days**	**Wed 14.01.09**	**Fri 30.01.09**	
2	Lese lignende oppgaver	10 days	Wed 14.01.09	Tue 27.01.09	
3	Lage webside	5 days	Mon 19.01.09	Fri 23.01.09	
4	Etablere KSE testmiljø	5 days	Mon 19.01.09	Fri 23.01.09	
5	Ferdig oppsatt testmiljø	0 days	Fri 23.01.09	Fri 23.01.09	4
6	Førsteutkast forprosjektrapport	0 days	Fri 23.01.09	Fri 23.01.09	
7	Møte veileder	0 days	Fri 23.01.09	Fri 23.01.09	
8	Lage outline til hovedrapport	5 days	Mon 26.01.09	Fri 30.01.09	
9	Redigering/Gjennomlesing av forprosjekt	5 days	Mon 26.01.09	Fri 30.01.09	
10	**Levere forprosjekt**	0 days	Fri 30.01.09	Fri 30.01.09	
11	Levere prosjektavtale	0 days	Wed 14.01.09	Wed 14.01.09	
12	**Hoveddel**	**86 days**	**Mon 26.01.09**	**Mon 25.05.09**	
13	Konseptuell fase	16 days	Mon 26.01.09	Fri 13.02.09	
14					
15	**Utarbeide rammeverk**	**45 days**	**Mon 09.02.09**	**Fri 10.04.09**	
16					
17	Datainnsamling	40 days	Mon 09.02.09	Fri 03.04.09	
18	Filer	10 days	Mon 09.02.09	Fri 20.02.09	
19	Prosesser	10 days	Mon 23.02.09	Fri 06.03.09	
20	Nettverk	10 days	Mon 09.03.09	Fri 20.03.09	
21	Register	10 days	Mon 23.03.09	Fri 03.04.09	
22	Minne	10 days	Mon 23.03.09	Fri 03.04.09	
23	Dataanalyse	30 days	Mon 02.03.09	Fri 10.04.09	
24	Presentasjon resultater	16 days	Fri 20.03.09	Fri 10.04.09	
25	**Fiktiv exploit**	**15 days**	**Mon 06.04.09**	**Fri 24.04.09**	
26	Datainnsamling	10 days	Mon 06.04.09	Fri 17.04.09	
27	Dataanalyse	10 days	Mon 06.04.09	Fri 17.04.09	
28	Policy og brukerveiledning	5 days	Mon 20.04.09	Fri 24.04.09	
29	**Kjent exploit**	**5 days**	**Mon 27.04.09**	**Fri 01.05.09**	
30	Datainnsamling	5 days	Mon 27.04.09	Fri 01.05.09	
31	Dataanalyse	5 days	Mon 27.04.09	Fri 01.05.09	
32	**Kompleks exploit**	**5 days**	**Mon 04.05.09**	**Fri 08.05.09**	
33	Datainnsamling	5 days	Mon 04.05.09	Fri 08.05.09	
34	Dataanalyse	5 days	Mon 04.05.09	Fri 08.05.09	
35	Ferdigstille rapport	8 days	Mon 11.05.09	Wed 20.05.09	
36	**Levere rapport kopisentral**	**0 days**	**Wed 20.05.09**	**Wed 20.05.09**	35
37	Levere rapport kopisentral	0 days	Mon 25.05.09	Mon 25.05.09	
38	**Plakat**	**8 days**	**Thu 21.05.09**	**Tue 02.06.09**	
39	Lage plakat	6 days	Thu 21.05.09	Thu 28.05.09	
40	Levere kopisentral	0 days	Thu 28.05.09	Thu 28.05.09	39
41	Levere studenttorget	0 days	Tue 02.06.09	Tue 02.06.09	
42	**Presentasjon**	**5 days**	**Wed 27.05.09**	**Wed 03.06.09**	
43	Forberede presentasjon	4 days	Wed 27.05.09	Mon 01.06.09	
44	Presentasjon	0 days	Wed 03.06.09	Wed 03.06.09	43
45					
46	**Møter**	**70 days**	**Fri 06.02.09**	**Thu 14.05.09**	
47	Møte veileder	0 days	Fri 06.02.09	Fri 06.02.09	
48	**Møte veileder og E&Y**	0 days	Fri 20.02.09	Fri 20.02.09	
49	Møte veileder	0 days	Fri 13.03.09	Fri 13.03.09	
50	Møte veileder	0 days	Fri 03.04.09	Fri 03.04.09	
51	**Møte veileder og E&Y**	0 days	Tue 28.04.09	Tue 28.04.09	
52	Møte veileder	1 day	Thu 14.05.09	Thu 14.05.09	

Møtereferat 16/12 2008

Tidspunkt: 09.00 – 13.30
Sted: Atrium

Til stede

- Hasse Kristiansen EY
- Eirik Tormodsrud EY
- Andre Årnes HiG
- Mathias Bjerke HiG
- Truls Hagen HiG
- Gudmund Norstrøm HiG
- Bjørnar Prestaasen HiG

Saksliste

Først hadde vi en oppgavegjennomgang der EY gikk igjennom hva de ønsket og få ut av oppgaven.
Videre diskuterte vi ulike utfordringer som ligger i oppgaven og vi satte ned flere punkter for å lettere finne en problemstilling.
Etter lunsj hadde vi brainstorming og laget et mindmap sammen med Eirik, Carsten og Sverre.
Etter brainstormingen skal vi lage en prosjektplan og lage en problemstilling for å kunne komme godt igang med prosjektet.

Møtereferat 06/02 2008

Tidspunkt: 13.00 – 13.30

Sted: HiG - telefonmøte

Til stede

- André Årnes HiG
- Mathias Bjerke HiG
- Truls Hagen HiG
- Gudmund Norstrøm HiG
- Bjørnar Prestaasen HiG

Saksliste

Først hadde vi en kort statusoppdatering av hvor vi er nå. Vi har igjen litt jobb på den konseptuelle fasen, men har kommet godt igang med å overvåke filer.

Deretter hadde vi en gjennomgang av disposisjonen til hovedrapporten. Vi hadde en diskusjon rundt metodikk, rammeverk og eksperiment og testmiljøkapittelet. Gruppen var litt usikre på innholdet i disse kapitlene og vi fikk gode tips fra veileder.

Prosjektavtalen ligger nå hos studiedirektør og vi venter på svar fra skolen.

Vi avtalte å ha et lite telefonmøte sammen med EY slik at de også får en innsikt i fremgangen i prosjektet.

Møtereferat 20/02 2008

Tidspunkt: 10.00 – 12.00

Sted: HiG - telefonmøte

Til stede

- Eirik Thormodsrud EY
- André Årnes HiG
- Mathias Bjerke HiG
- Truls Hagen HiG
- Gudmund Norstrøm HiG
- Bjørnar Prestaasen HiG

Saksliste

Møtet startet med en statusoppdatering der gruppa forklarte i korte trekk hva vi har gjort frem til nå og hvordan vi skal jobbe videre med prosjektet. Gruppa har kommet godt i gang og ligger godt an i forhold til tidsplanen vi har satt opp.

Deretter gikk vi igjennom hvert kapittel i rapporten og diskuterte endringer og oppsett av disse. Vi fikk gode tilbakemeldinger på hvordan rapporten så ut og hadde fått med mye interessant stoff. Oppsettet i kapittel 4 ble også endret da vi var litt usikre på hva dette skulle inneholde.

Møtet ble avsluttet med en demo der vi overvåket Windows og utførte noen filendringer som ble oppdaget av procmon, og ble hentet ut i en leselig rapport. Det ble også tid til en lunsj i kantina til slutt.

Deretter diskuterte vi litt rundt møtet vi skal ha den 20/4.

Sammendrag prosjektdagbok

Prosjektets arbeidstid er bestemt i gruppereglene til å være fra 8.00 til 15.00. På mandager har tre av gruppemedlemmene et tilleggsfag, IT Service Management, slik at på mandager er det avsatt fire timer til prosjektarbeid. Det ukentlige timeantallet blir dermed på 32 timer per uke.

Totalt antall arbeidstimer i gruppa blir 2304. Gjennomsnittlig arbeidstimer per gruppemedlem blir 576. Det har ikke blitt lagt inn kveld- og helgejobbing i utregningen slik at gjennomsnittlig timeantall øker noe for hvert gruppemedlem.

Videre kommer et sammendrag over loggbok-funksjonen som ligger i prosjektstyringsverktøyet.

Onsdag 14. januar 2009

Bjørnar

- Laget forslag til møteplan (sendt Andre og EY)
- Mail prosjektoppgave Natasja
- Lagt inn prosjektorganisering i prosjektplan
- lastet ned MS Project og startet på gantt diagram

Gudmund

- Satt meg inn i subversion
- Laget kapittel-filer for forprosjektrapport i latex
- Satte opp mal for forprosjekt i latex med innholdsfortegnelse.

Mathias

- Fordele roller
- Sette seg inn i trac
- Legge opp oppgaver på trac
- Administrativt

Truls

- Sette seg inn i subversion og opprette maler for forprosjektet
- Skrevet i forprosjektrapporten
- Lese tidligere prosjekter

Torsdag 15. januar 2009

Bjørnar

- Laget utkast til gantt-skjema
- Laget nytt utkast tl problemstilling og sendt til veileder
- Sendt prosjektavtale til EY
- Jobbet med forprosjektrapporten

Gudmund

- Grovt gantt-diagram
- Forside til rapporter i latex
- Jobbet med forprosjekt.

Mathias

- Etterlyse utstyr fra it-tjeneste
- Jobbe med problemstilling
- Jobbe med forprosjektrapport
- Fordele oppgaver

Truls

- Satt opp kontrakt mellom HiG og oppdragsgiver
- Jobbet med forprosjekt
- Lest tidligere prosjektoppgaver