



Denne forelesningsøkten vil bli tatt opp og lagt ut i emnet i etterkant.

Hvis du ikke vil være med på opptaket:

 Start Video	La være å delta med webkameraet ditt.
 Unmute	La være å delta med mikrofonen din.
To: Marianne Sundby (Privately) Type message here...	Still spørsmål i Chat i stedet for som lyd. Hvis du ønsker kan spørsmålet også sendes privat til foreleser.



Høyskolen
Kristiania

TK2100: Informasjonsikkerhet

Femte forelesning:

MALWARE MED KINETISK UTFALL

Innside-angrep

- Innside-angrep
 - skyldes/tilrettelegges av noen som er del av organisasjonen som kontrollerer eller bygger tjenesten som skulle vært beskyttet
 - «Utro tjenere»
- En **bakdør** er en skjult metode som (typisk) tillater en bruker å utføre handlinger han normalt ikke har tillatelse til
- **Logikkbomber** utfører en handling først når en bestemt **betingelse** inntreffer

Forsvar mot ^(Oppsummering)inside-angrep

- Unngå «single points of failure»
- Bruke (manuell) kode-gjennomgang
- Bruk arkiveringsverktøy og rapport-verktøy
- Begrens tillatelser og autorisasjoner
- Fysisk sikring av kritiske systemer
- Overvåk ansattes adferd
- Stålkontroll på alt som installeres

- Eller: Open Source (ref: Silent Circle)

Klassifikasjon («historisk»)^(Oppsummering)

- Vi kan dele **malware** opp i ulike **typer** ut fra hvordan den spres og hvordan den skjuler seg.
- Spredning
 - **Virus**: Virus endrer eksisterende filer eller systemer, koden kan ikke leve eller spre seg alene
 - **Orm**: automatisk spredning fra maskin til maskin over nettet
- Skjuler seg
 - **Rootkit**: endrer OS for skjule nærvær
 - **Trojaner**: Nytteprogram som skjuler ondsinnede operasjoner (f.eks. keylogger)
- «Nyttelast» (payload)
 - Alt fra humor/irritasjon til ran av maskinkraft og identitetstyveri

Hva er et computer virus?

- Et program som kan **replisere** seg selv
 - ved å endre andre filer/program
 - ved å **infisere** dem med kode
 - som kan **formere** seg videre
- Det er evnene til å **formere seg LOKALT** som skiller virus fra andre typer malware
- Krever vanligvis innledende **brukermedvirkning** for å formere seg
 - Klikke på en link og godta installasjon
 - Åpne epost-vedlegg
 - Dele en **minnepinne**, eller annet USB-utsyr

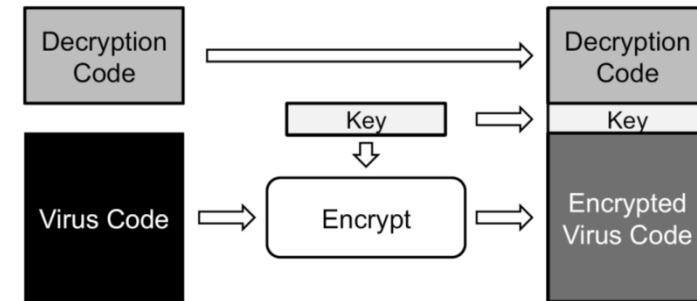
Tradisjonelle datavirus

(Oppsummering)

- I dag er det mer vanlig med ormer og trojanere; malicious filer som er i stand til å leve et selvstendig liv uten en host-prosess
- Ormer har eksistert lengre enn PCer, og første registrerte malware kom i 1971 og kalles "The Creeper Program" og spredde seg over ArpaNet
- Brain krediteres som verdens første "PC virus" (1986), og senere samme år klarte man for første gang å infisere exe filer med Suriv-02
 - Exe filer var først ansett som et trygt format fordi det var så kompleks at ingen ville kunne klare å infisere dem, i motsetning til com filer som er ren maskinkode...
 - Noen krediterer Old Yankee som den første exe fil infektoren
- Flere farlige virus kom ut på denne tiden:
 - AIDS Trojan (1989); krypterte hele disken din
 - Dark Avenger (1989); overskrev random deler av disken 1/16 ganger viruset kjørte
 - Jerusalem (1987); Sletter filer på maskinen på Fredag 13nde...
 - Tequila (1991); Polymorph virus som var stealthet, og veldig vanskelig å oppdage

Teknikker for å gjemme seg (Oppsummering)

- **Krypterte** virus
 - **Dekrypteringsmotor** + kryptert virus
 - Tilfeldig generert krypteringsnøkkel
 - Antivirus søker etter dekrypteringsmotoren
 - Ofte bare for å skjule koden, men kan gi:
- **Polymorfe** virus
 - Virus legger inn tilfeldige variasjoner i koden sin før den sprer seg videre
 - En polymorphic engine trengs for å dekode viruset før det kan kjøre
 - Kan detekteres med CPU-emulator
 - Må finne signatur basert på evnen til å endre seg selv
- **Metamorfe** virus
 - Forsøker å gjemme seg og være vanskelige å finne en signatur på ved «obskurifisering»:
 - endre rekkefølgen på instruksjoner
 - Legge inn unyttige instruksjoner
 - Omstrukturere indre metode-kall
 - Kan bruke statistiske metoder for å finne sannsynlige virus ut fra et bestemt antall under-signaturer



- Å gjemme seg for anti-virus handler da ofte om å **skjule signaturen**, eller gjøre den vanskelig å generere

Hva er en orm («Worm»)?^(Oppsummering)

- Malware som **sprer kopier** av seg selv **uten å infisere** andre program, og vanligvis uten menneskelig medvirkning
- Ikke virus siden de ikke infiserer eller endrer LOKALT
 - men begge deler spres ved selv-replisering
- I de fleste tilfeller vil ormen ha en ondsinnet **nyttelast (payload)**
 - Installere bakdør
 - Slette filer

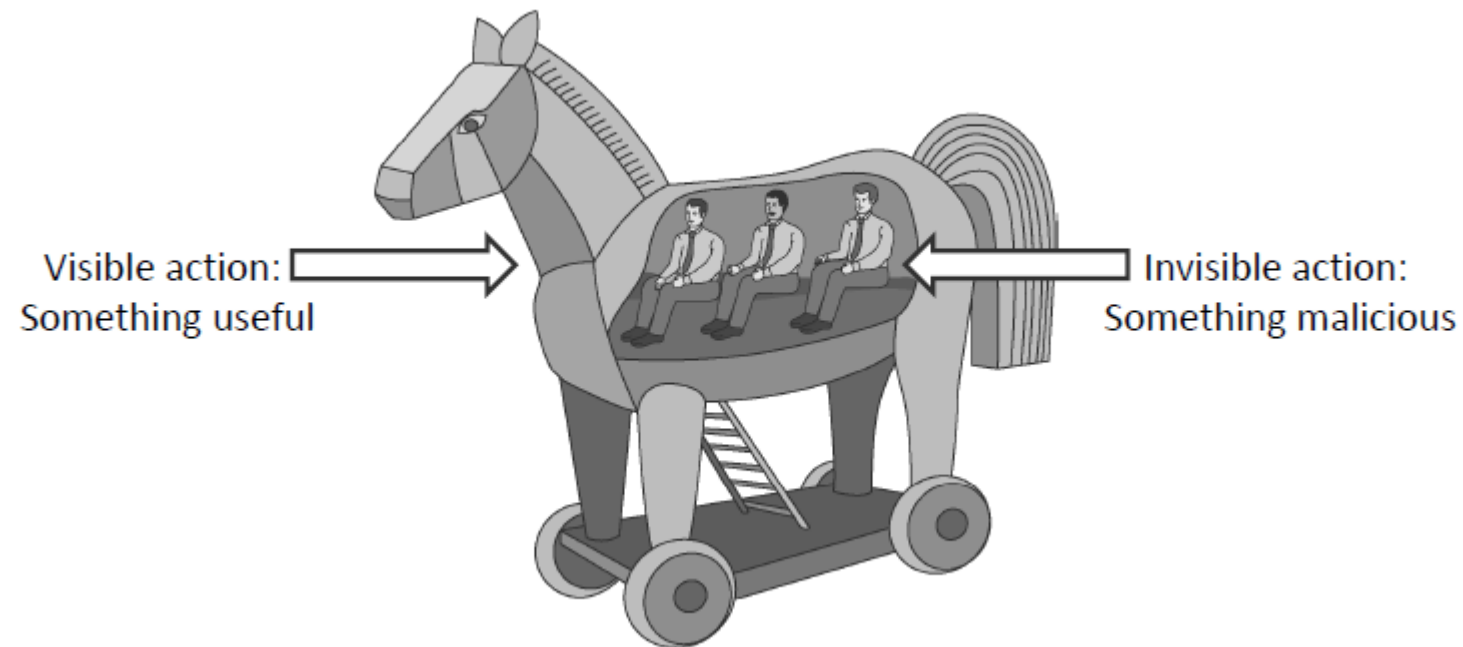
Mass mailers, ormer

(Oppsummering)

- 4. mars 2000 kom 'ILOVEYOU' ormen – og endret "alt"
 - Den første malware som spredde seg gjennom epost
 - Brukeren måtte manuelt åpne en fil i eposten
 - Egentlig ikke en skadelig malware, den bare spredde seg...
 - 50 millioner PCer ble infisert i løpet av 9 dager!
- 13. Juli 2001 infiserte 'Code Red' Microsoft IIS servere
 - Helt ny orm som spredde seg automatisk mellom servere på Internet, og som kun levde inne i Internet Information Server
 - Utførte Denial of Service attacks mot flere amerikanske nettsteder
- 'Nimda' ormen fulgte opp og satt standarden for moderne malware
 - Spredning gjennom; epost, nettverks shares, IIS spredning (som Code Red), browsing på infiserte servere, gjennom bakdører fra andre ormer

Trojanske hester

- Malware som **ser ut** til å utføre en **nyttig** jobb, men som **i tillegg** gjør noe ondsinnet
 - F.ex. starter en keylogger («tastaturavlytter»)
- Trojanere installeres ofte som en del av nyttelasten til annen malware
 - men kan også installeres av bruker/administrator med overlegg eller ved uhell



Rootkits

- Rootkit modifierer **OS** og **skjuler** sin eksistens
 - F.ex. Modifiserer filsystemets muligheter til å se en fil
 - Vanskelig å oppdage med programvare som jo er avhengig av OSet selv
- Fra 'Operativsystem sikkerhet':
 - Rootkits i BIOS
 - Rootkits i CPU (Blue Pill rootkit'et)



Banktrojanere

- Zeus / GameOver dukket opp i 2007, men fikk mye oppmerksomhet først i 2009 – den stjal passord til flere nettsteder; blant annet Bank of America
- Flere rettede trojanere dukket opp i denne perioden, også flere som singlet ut norske bankkunder
- Phishing og pharming er vanlig, hvor man leder brukeren til en side som ser ut som bankens innloggingsside og ber om passord
- I 2010 dukker SpyEye opp, et "klikk-og-dra" programmeringsverktøy for å lage malware som stjeler bankinformasjon.
 - Det lages rettede angrep mot Nordea og DnB NOR
 - Norske banker går ut i media februar 2011 og advarer kundene
 - SpyEye koster 7000 kroner og selges til alle som ønsker å stjele bankinformasjon slik at disse kan lage trojanere uten særlig kunnskap...
 - Til tross for høyt sikkerhetsnivå i norske banker ble flere svindlet

- Et sikkerhetsfirma i Hviterussland; VirusBlokAda, fikk en epost fra et iransk firma som hadde problemer med at servere rebootet, og de fryktet at de var infisert med et ukjent datavirus
- VirusBlokAda, med Sergey Ulasen i spissen, jobbet med maskinene i en uke, og ble til slutt sikre på at de hadde funnet et rootkit på flere av serverne – de varslet da resten av sikkerhetsindustrien
- De mistenksomme filene som ble funnet var ukjente drivere (.sys filer) som var digitalt signert med Realtek sitt private sertifikat!
- "Viruset" hadde smittet maskiner som ikke hadde internettilgang
- Infeksjonsvektoren hadde vært smittede USB sticker, og det virket ikke som at viruset hadde evne til å smitte videre...
- Ut ifra fildatoer og annen informasjon ble flere av maskinene infisert i juni 2009 – altså ett år før det ble oppdaget

- Hvorfor var infeksjon gjennom USB stikker så alarmerende?
- Det hadde aldri før skjedd at malware var signert av et ekte firma sitt sertifikat – men er dette alarmerende nok til å holde en forelesning om? :-)
- Det var ikke infeksjonsvektoren som var alarmerende – det var hva som ble smittet som var alarmerende...
- Det var nemlig ikke vanlige servere som var smittet...
- Maskinene kjørte som SCADA PLC kontroll maskiner...
- Som styrte:
 - Atomreaktorer!



Neida, det
eksploderte
ikke – vi
kommer tilbake
til hva som
skjedde :-)

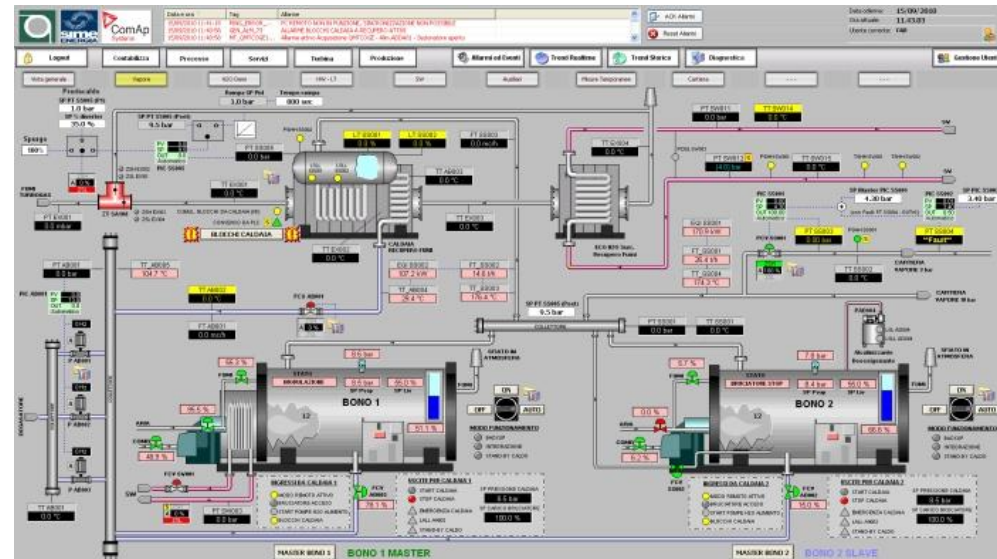
- 15. juli 2010 skriver Brian Krebs en blog om StuxNet, og det blir allment kjent at StuxNet eksisterer
- Samme dag utføres et Denial of Service Attack mot flere mailinglister og websider, hvilket gjør det ekstra vanskelig for researchere, og drifts administratorer for SCADA nettverk, å få tilgang til informasjon
- 15. juli 2010 Norman detekterer StuxNet med sin virussøkemotor

- Microsoft offentliggjør en rapport CVE-2010-2568, som beskriver en exploit i LNK formatet som StuxNet utnytter for å infisere maskiner automatisk når man setter inn en USB stick i en Windows maskin
- 19. juli 2010 Hackere/researchere offentliggjorde LNK exploit koden på internett, med guide for hvordan man skulle bruke det i annen malware
- Microsoft går ut med en statement om at CVE-2010-2568 ikke er en bug, men «as designed» og ikke vil bli fikset (for å ikke knekke eksisterende, lovlig programvare)
- 28. juli 2010 Norman sine produkter stopper generisk varianter av exploiten
- 02. august 2010 Microsoft går tilbake på sitt opprinnelige standpunkt – og releaser MS10-046 som stopper LNK exploiten på de systemene som blir patchet

- Hovedvektor var gjennom en exploit av LNK fil formatet
- Infiserte gjennom exploits videre i LANet (RPC og Print Spooler exploits)
- Totalt ble 4 ukjente "zero day" exploits brukt!
- Har en innebygget P2P mekanisme, brukes blant annet for å oppdatere seg
- Ville bare spre seg videre til noen få maskiner før den ble uvirkesom, for å ikke bli oppdaget
- Maskiner som kjørte Siemens SCADA systemer ble grundig infisert
- Viruset sprer seg videre fra SCADA maskinen til Siemens sine PLC kontrollere gjennom en feil i WinCC databasen
 - Hoved administrator passordet for PLCen var likt på alle installasjoner
 - En bug i SCADA klienten gjorde at hvis man forsøkte å endre passordet så krasjet klienten, Siemens visste ikke om denne feilen – ergo er det INGEN som bytter dette passordet...
 - StuxNet viruset hadde dette passordet hardkodet :-)

Hva er SCADA kontroll systemer?

- Siemens SCADA systemer brukes for å kontrollere mange typer systemer hvor mennesker og maskiner samarbeider
- Industrielle prosesser
 - Fabrikk maskiner
 - Kraftverk
 - Oljerafinerier
- Infrastrukturelle prosesser
 - Vannrenseverk
 - Olje- og gassledninger
 - Strømtilførsel
- Drift av "facilities"
 - Maritim skipsfart
 - Romfart
 - Flyplasser
 - Andre større bygninger



- StuxNet infiserte kun systemet hvis følgende var oppfylt:
 - Siemens S7-300
 - Variable frequency drive fra Vacon (Finland) eller Fararo Paya (Iran)
 - Kun motorer med frekvens mellom 807 Hz og 1210 Hz
- Som visstnok er akkurat det oppsettet atomanlegg i Iran har!
- StuxNet sendte data ut til to websider (en i Danmark og en i Malaysia) om data den fant i SCADA systemet
- StuxNet hadde også kode for å justere roteringshastigheten på motorene koblet til systemet – og å skjule disse endringene fra monitor systemene!

Land hvor StuxNet ble funnet

- Land hvor StuxNet ble oppdaget *

- Iran (~ 60%)
 - Merk at alle PRIMÆR infeksjoner var i Iran
 - Indonesia
 - India
 - Equador
 - Pakistan
 - Libanon
 - Taiwan
 - Azerbaijan
-
- *) Symantec sin statistikk, og Microsoft Malware Protection Center

LNK exploitet i mer detalj

- Infiserte USB stikker ville automatisk infisere en Windows maskin når den ble plugget inn i maskinen, når Windows forsøkte å laste et ikon for USB stikken
- .LNK filer er filer du typisk finner på Windows Desktop eller Start Menu, og er kun linker til andre filer
- LNK filformatet er et binært format som er bygget opp som:
 - `SHELL_LINK = SHELL_LINK_HEADER`
`[LINKTARGET_IDLIST]`
 - `[LINKINFO] [STRING_DATA]`
`*EXTRA_DATA`
 - `SHELL_LINK_HEADER` inneholder timestamps og diverse flagg, som ikke er relevant for exploitet.

LNK exploitet i mer detalj #2

- LINKTARGET_IDLIST inneholder en optional struktur som angir filplassering på disk.

- ```
typedef struct _IDLIST {
```
- ```
    ITEMID aItemID[1]; // [variable];
```
- ```
 UINT16 usTerminalID;
```
- ```
} IDLIST, *PIDLIST;
```

- ```
typedef struct _ITEMID {
```
- ```
    UINT16 usItemIDSize;
```
- ```
 BYTE Data[1];
```
- ```
} ITEMID, *PITEMID;
```


LNK exploitet i mer detalj #3

- LNK filene som ble funnet ifm StuxNet viruset så slik ut:

- ```
IDListSize = 36 01
idlist
Item1 (20 bytes)
 ItemIDSize = 14 00
 Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2 D8 08 00 2B 30 30 9D
Item2 (20 bytes)
 ItemIDSize = 14 00
 Itemid = 2E 1E 20 20 EC 21 EA 3A 69 10 A2 DD 08 00 2B 30 30 9D
Item3 (268 bytes)
 ItemIDSize = 0C 01
 Itemid = 00 00 EXTRA_DATA TerminalBlock = 00 00 00 00
```

# LNK exploitet i mer detalj #4

- ```
Item1 ( 20 bytes )
ItemIDSize = 14 00
Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2
D8 08 00 2B 30 30 9D
```

- Item ID 1F angir en CLSID tag, som er $0x14 = 20$ bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
```
- ```
    UINT16 usUnknown;
```
- ```
 CHAR8 acClsid[1];
```
- ```
} ITEMID_CLSIDELEMENT, *PITEMID_CLSIDELEMENT;
```

- {20D04FE0-3AEA-1069-A2D8-08002B30309D}

LNK exploitet i mer detalj #4

- ```
Item1 (20 bytes)
ItemIDSize = 14 00
Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2
D8 08 00 2B 30 30 9D
```

- Item ID 1F angir en CLSID tag, som er  $0x14 = 20$  bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
```
- ```
 UINT16 usUnknown;
```
- ```
    CHAR8 acClsid[1];
```
- ```
} ITEMID_CLSIDELEMENT, *PITEMID_CLSIDELEMENT;
```

- {20D04FE0-3AEA-1069-A2D8-08002B30309D}

# LNK exploitet i mer detalj #4

- ```
Item1 ( 20 bytes )
ItemIDSize = 14 00
Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2
D8 08 00 2B 30 30 9D
```

- Item ID 1F angir en CLSID tag, som er $0x14 = 20$ bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
```
- ```
    UINT16 usUnknown;
```
- ```
 CHAR8 acClsid[1];
```
- ```
} ITEMID_CLSIDELEMENT, *PITEMID_CLSIDELEMENT;
```

- {20D04FE0-3AEA-1069-A2D8-08002B30309D}

LNK exploitet i mer detalj #4

- ```
Item1 (20 bytes)
ItemIDSize = 14 00
Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2
D8 08 00 2B 30 30 9D
```

- Item ID 1F angir en CLSID tag, som er  $0x14 = 20$  bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
    UINT16 usUnknown;
    CHAR8 acClsid[1];
} ITEMID_CLSIDELEMENT, *PITEMID_CLSIDELEMENT;
```

- {20D04FE0-3AEA-1069-A2D8-08002B30309D}

LNK exploitet i mer detalj #4

- ```
Item1 (20 bytes)
ItemIDSize = 14 00
Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2
D8 08 00 2B 30 30 9D
```

- Item ID 1F angir en CLSID tag, som er  $0x14 = 20$  bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
```
- ```
 UINT16 usUnknown;
```
- ```
    CHAR8 acClsid[1];
```
- ```
} ITEMID_CLSIDELEMENT, *PITEMID_CLSIDELEMENT;
```

- {20D04FE0-3AEA-1069-A2D8-08002B30309D}

# LNK exploitet i mer detalj #4

- ```
Item1 ( 20 bytes )
ItemIDSize = 14 00
Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2
D8 08 00 2B 30 30 9D
```

- Item ID 1F angir en CLSID tag, som er $0x14 = 20$ bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
```
- ```
    UINT16 usUnknown;
```
- ```
 CHAR8 acClsid[1];
```
- ```
} ITEMID_CLSIDELEMENT, *PITEMID_CLSIDELEMENT;
```

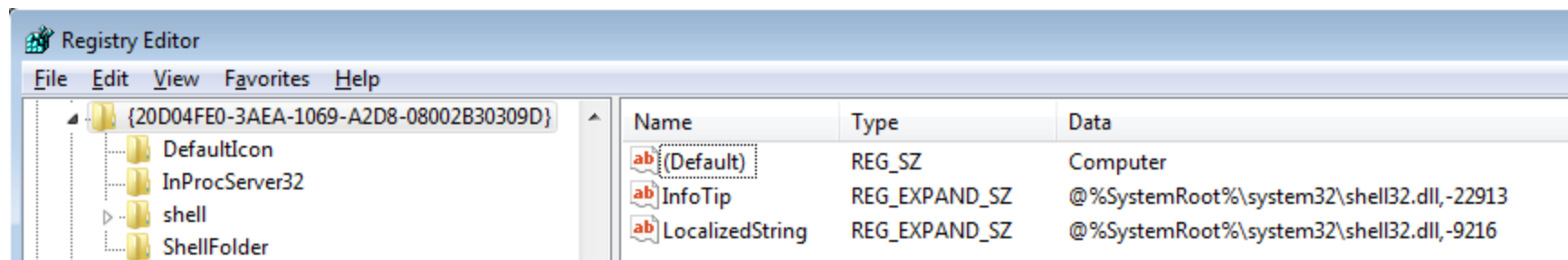
- {20D04FE0-3AEA-1069-A2D8-08002B30309D}

LNK exploitet i mer detalj #4

- ```
Item1 (20 bytes)
ItemIDSize = 14 00
Itemid = 1F 50 E0 4F D0 20 EA 3A 69 10 A2
D8 08 00 2B 30 30 9D
```

- Item ID 1F angir en CLSID tag, som er  $0x14 = 20$  bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
```
- ```
 UINT16 usUnknown;
```
- ```
    CHAR8 acClsid[1];
```

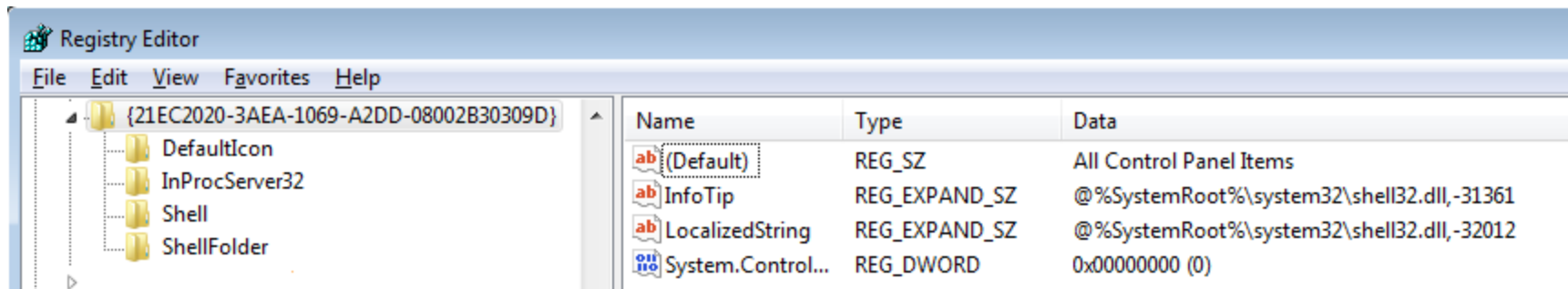


LNK exploitet i mer detalj #5

- ```
Item2 (20 bytes)
ItemIDSize = 14 00
Itemid = 2E 1E 20 20 EC 21 EA 3A 69 10 A2
DD 08 00 2B 30 30 9D
```

- Item ID 2E angir en annen CLSID tag, som er 0x14 = 20 bytes stor

- ```
typedef struct _ITEMID_CLSIDELEMENT {
```
- ```
 UINT16 usUnknown;
```
- ```
    CHAR8 acClsid[1];
```



LNK exploitet i mer detalj #6

- ```
Item3 (268 bytes)
ItemIDSize = 0C 01
Itemid = 00 00 EXTRA_DATA
```
- Item ID 00 var spesiell, det er mest sannsynlig en gjenglemt debug verdi som var laget av Microsoft sine utviklere når de laget .lnk formatet, og var aldri ment brukt. Det var en forenkling av path objekter, og angir enkelt og greit en komplett path til filen som skal brukes.
- ```
typedef struct _STRING_DATA {
    UINT16 usCountCharacters;
    WCHAR awString[1];
} STRING_DATA, *PSTRING_DATA;
```
- Og her var exploiten, hvis det forrige elementet henviste til Control Panel, og hadde den "gjenglemte" Item ID 00 så ble ikonet lastet feil. – Mer teknisk:

- LoadLibrary

- [http://msdn.microsoft.com/en-us/library/windows/desktop/ms684175\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms684175(v=vs.85).aspx)
- If the specified module is a DLL that is not already loaded for the calling process, the system **calls the DLL's DllMain function with the DLL_PROCESS_ATTACH value**. If DllMain returns TRUE, LoadLibrary returns a handle to the module. If DllMain returns FALSE, the system unloads the DLL from the process address space and LoadLibrary returns NULL.

- LoadLibraryEx

- [http://msdn.microsoft.com/en-us/library/windows/desktop/ms684179\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms684179(v=vs.85).aspx)
- LOAD_LIBRARY_AS_DATAFILE
- 0x00000002
- If this value is used, the system maps the file into the calling process's virtual address space as if it were a data file. **Nothing is done to execute or prepare to execute** the mapped file. Use this flag **when you want to load a DLL only to extract messages or resources** from it.

LNK exploitet i detalj – forklart

- På grunn av at Windows Shell, for en spesifikk – ikke dokumentert, Item ID brukte LoadLibrary og ikke LoadLibraryEx ble kode i DLLen kjørt
- Dette ble utnyttet til å kjøre en executable på en USB pinne automatisk
- Microsoft anså allikevel ikke dette som et exploit, men "as designed", men etter en del press fra sikkerhetsbransjen releaset de en patch som endret dette etter cirka en måned
- Dette var en ukjent exploit (dvs ingen andre malware brukte det) frem til StuxNet ble avslørt
- Windows 98, Windows Me, Windows NT 4.0, Windows XP, Windows Vista og Windows 7 var sårbare da det var en core komponent som ikke var blitt endret siden den ble laget for første gang

- **Cofer Black** (2011, Black Hat Las Vegas)
- 'Head of Counter Terrorism', CIA (1999 – 2002)
- Until recently, the U.S. Government counterterrorism groups have been focused on the possibility of chemical, bacteriological, radiological and nuclear attacks. The appearance of Stuxnet has changed that, and the concerns are now kinetic, bacteriological and cyber.
- I am here to tell you, and you can quote me on this: The Stuxnet attack is the Rubicon of our future. It was really expensive, so a nation-state had to be involved. Second, [the world of cyber] has now morphed into physical destruction of national resources. This is huge.
- This brings in viability of kinetic counterstrike.

Hva kan gjøres med slik malware?

- Åpenbart; ødelegge komponenter i kjernekraftverk
 - Rundt 50% av alle maskiner hvor StuxNet ble funnet er i følge en uttalelse fra forskeren Ralph Langner på to kjernekraftverk i Iran.
 - I følge en rapport fra Institute of Science and International Security ble 1000 sentrifuger ødelagt ved Natanz kraftverket mellom November 2009 og Januar 2010
 - StuxNet var spesifikt skrevet for å først øke roteringshastigheten i motoren, for så å senke den kraftig – hvilket vil resultere i vibrasjoner som resulterte i "forurenset" uran, og skadet sentrifugen
- Endre temperaturen i kraftverk – og resultere i meltdown/eksplosjon?
- Koble ned strømtilførselen i et land
- Industrispionasje
- Ødelegge fabrikkene til et konkurrerende firma

- I juni 2012 avslørte Washington Post det som mange hadde mistenkt
- Stuxnet var utviklet av NSA, CIA og Israelsk etterretning
 - Den amerikanske avdelingen som stod bak mesteparten av koden og forskningen har ikke noe offisielt navn.
 - Den israelske avdelingen kalles enten Unit 2600, eller Unit 8200 - avhengig av kilde (<https://twitter.com/y0m/status/126991414516121600>)
- Operasjonen ble godkjent av Bush administrasjonen I 2006
- Viruset var laget for å infisere anlegg for anriking av uran I Iran
- Så får hver enkelt mene sitt om personer som er involvert i så hemmelige militære operasjoner "lekker" informasjon til pressen, eller om dette er kontrollert informasjon som amerikanske myndigheter vil at alle skal få vite om – en del av det som ble "avslørt" stemmer nemlig ikke...

Op Olympic Games - Natanz

- Januar 2010 oppdager inspektører fra Atomic Energy Agency at sentrifuger ved anlegget blir ødelagt i et tempo uten sidestykke
- De iranske forskerne forstår ikke hvorfor, alt deres utstyr tilsier at de gjør alt korrekt
- 5 måneder tar det altså før StuxNet blir avslørt på serverne, og de blir avdekket på grunn av noe helt urelatert...
- Ingen, i sin fjerneste fantasi, kunne tro at et «datavirus» kunne forårsake fysisk ødeleggelse på atom-sentrifuger!



Strategiske mål for operasjonen

- Amerikansk etterretning antok at Iran forsøkte å utvikle atomvåpen
- Hvis de hadde angrepet anlegg med våpen (eller offensive cyber-våpen) ville de bare startet forsøkene sine på nytt - hvis ikke USA erobret landet
- Ved å utvikle et virus som resulterte i "forurensset" uran håpet de at iranske myndigheter ville tro at forskerne deres ikke var dyktige nok – og dermed legge ned all forskning
 - Mistanke om egen inkompetanse var ansett som bedre enn ødeleggelse!
- Israel ble invitert med i operasjonen for å forhindre et preventivt rakett angrep mot Natanz fra Israel sin side
- Første del av operasjonen var å bruke forskjellige virus med beacon payloads til å finne ut hvordan Natanz var bygget opp og hva slags hardware som ble brukt
- Så ble en kopi av anlegget bygget opp i USA, og det ble forsket på hvordan de kunne sabotere forskningen – Stuxnet ble utviklet

Data man ikke leser om i avisene...

- Etter “avsløringene” i Washington Post og senere andre aviser virker det som at det kun var Natanz som ble angrepet av operasjonen, og alt annet var en feil i softwaren som gjorde at viruset rømte...
- Ut ifra virus samples som er analysert, viruset lagrer nemlig IP adresse for hver maskin den har infisert inne i seg selv, så stemmer nok ikke det
- Hele 5 forskjellige institusjoner i Iran ble utsatt for infeksjon, 3 forskjellige versjoner er identifisert, og 5 tilfeller av infeksjoner; i juni 2009, juli 2009, mars 2010, april 2010 og mai 2010
- Viruset var laget for å spre seg innover i nettet, men naturligvis vil et virus spre seg alle retninger – også ut av målet
- Totalt er 12.000 infeksjoner blitt identifisert, med over 100.000 infiserte maskiner i forskjellige nettverk
- Utilsiktet, eller akseptabel “collateral damage”?

Fra de ”lekkede” intervjuene

- Sitatet som fikk avisene til å tro (og rapportere) at KUN Natanz var målet:

“We think there was a modification done by the Israelis,” one of the briefers told the president, “and we don’t know if we were part of that activity.”

Mr. Obama, according to officials in the room, asked a series of questions, fearful that the code could do damage outside the plant. The answers came back in hedged terms. Mr. Biden fumed. “It’s got to be the Israelis,” he said. “They went too far.”

- Men vi vet som sagt at det var 5 rot infeksjoner på 3 forskjellige steder i Iran – så informantens utsagt er ”tilpasset” hva CIA og NSA vil at vi skal tro ;-)

- StuxNet ble funnet og fjernet – var ”vi” da trygge?
- 1. september 2011 ble et nytt virus oppdaget; Duqu som virker som at det oppfyller det samme formålet som StuxNet...
- Duqu bruker samme rootkit som StuxNet, men den bug'en som gjorde at servere krasjet så ofte – den er fikset :-)
- SCADA miljøet har fått en tankevekker, og Siemens og Emerson starter nå å forbrede sine systemer med sikkerhet i fokus
- Hvor omfattende cyber angrep som dette kan bli vil vi ikke vite før det neste angrepet kommer...
 - Det er spennende å jobbe i databransjen! :-)

Koden som overrasket alle

- En del av Duqu sitt rootkit ble en hard nøtt å knekke
- Researchere forstod ikke hva slags compiler som kunne lage slik kode – ei heller hvilket programmeringspråk dette var!

```
class2_ctor    proc near                ; CODE XREF: ...
arg_0_p_compare_func= dword ptr 4

    push     esi
    push     450h                       ; dwBytes
    call     new
    mov      esi, eax
    pop      ecx
    test     esi, esi
    jz       short loc_100125B3
    lea      eax, [esi+class_2.csec]
    push     eax                         ; lpCriticalSection
    call     ds:InitializeCriticalSection
    mov      eax, [esp+4+arg_0_p_compare_func]
    mov      [esi+class_2.setup_class13], offset class2_setup_class13
    mov      [esi+class_2.append], offset append_to_existing
    mov      [esi+class_2.remove], offset class2_remove ; (this, key)
    mov      [esi+class_2.clear], offset class2_clear
    mov      [esi+class_2.exists], offset class2_exists
    mov      [esi+class_2.count], offset class2_count
    mov      [esi+class_2.get_next_value], offset class2_get_next_value
    mov      [esi+class_2.get_prev_value], offset class2_get_prev_value
    mov      [esi+class_2.get_values_as_array], offset class2_get_values_in_array
    mov      [esi+class_2.dtor], offset class2_dtor
    mov      [esi+class_2.p_compare_func], eax
    call     class2_allocate_block_pair ; 1 = success
                                           ; 0 = fail
    test     eax, eax
    jnz      short loc_100125B7
    push     esi                         ; lpMem
    call     class2_dtor
    pop      ecx

loc_100125B3:                                     ; CODE XREF: ...
    xor      eax, eax
    pop      esi
    retn

; -----
loc_100125B7:                                     ; CODE XREF: ...
    mov      eax, esi
    pop      esi
    retn
class2_ctor    endp
```

”Old school” programmerere

- Det viser seg at denne koden er skrevet i en obskur dialekt av C, hvor utviklerne har brukt sitt eget C-påbygg for lage ”Object Oriented C” (altså ikke C++, og ikke Objective-C)
- Visual C 2008 med /OI /Obl, og et custom OOC bibliotek/framework

The use of object-oriented C to write the event-driven code in DuQu reveals something about the programmers who coded this part of DuQu – they were probably old-school coders (Kaspersky’s researchers)

The programming style is uncommon for malware and is more commonly found in professionally-produced commercial software created ten years ago

Kinetic counterstrike

- Vi har hørt Cofer Black sine antagelser om hva som kan skje
- Rapporten 'International Strategy for Cyberspace' sier:

When warranted, the United States will respond to hostile acts in cyberspace as we would to any other threat to our country. All states possess an inherent right to self-defense, and we recognize that certain hostile acts conducted through cyberspace could compel actions under the commitments we have with our military treaty partners. We reserve the right to use all necessary means — diplomatic, informational, military, and economic—as appropriate and consistent with applicable international law.

https://obamawhitehouse.archives.gov/sites/default/files/rss_viewer/international_strategy_for_cyberspace.pdf

- Oppdaget 28 mai 2012 av den iranske CERT organisasjonen
- Kun 1000 infiserte maskiner...
 - Primært i Iran, Lebanon, Sudan, Syria, Saudi Arabia og Egypt
- Ren spion programvare for klient maskiner, med mulighet til å:
 - Ta screenshots på maskinen
 - Ta opp audio via maskinens mikrofon
 - Key logger som lagrer alt som blir skrevet på tastaturet
 - Logger all inn- og utgående nettverkstrafikk
 - Tar opp Skype samtaler
 - Sender dokumenter og AutoCAD tegninger
- Tidenes mest omfattende malware – på hele 20 MB
- 8 juni 2012 (11 dager etter den ble oppdaget) begynte Command and Control sentrene til malwaren og sende "kill" kommandoer som slettet alle spor etter viruset på infiserte maskiner!
- I følge Washington Post avsløringene – en del av Olympic Games

Og hva skjedde i 2017?

- Vault 7; 1000 hacker verktøy utviklet av CIA og NSA lekket ut på wikileaks...
- <https://www.wikileaks.com/ciav7p1/>
- Foreløpig ikke noe kildekode, men omfanget er grundig dokumentert



Et lite utdrag fra "Vault 7"

- Weeping Angel; avlytting av Samsung smart-tv (laget sammen med MI5)
- Remote control av biler (snikmyrding ved å kjøre de av veien)
- Zero-day angrep mot iPhone, Android, Windows, Max OS X, Solaris, Linux – hvis det kjører et OS har/hadde CIA et hack for å infisere eller ta kontroll...
- Fine Dining; trojaner suite som gir inntrykk av at agenten bare spiller av en film el

Nå – også kommersielt

- I 2021 har den store «snakkis'en» i sikkerhetsmiljøet vært NSO Group
- Israelsk «cyber våpen» selskap som besitter zero-day angrep mot både Android og Apple som kan fjerninstallere overvåkningsprogramvare
- Solgte spionpakken PEGASUS til alle verdens myndigheter, også land som har brukt dette i mindre demokratisk øyemed...

[https://en.wikipedia.org/wiki/Pegasus_\(spyware\)](https://en.wikipedia.org/wiki/Pegasus_(spyware))

<https://www.theverge.com/2021/12/21/22848485/pegasus-spyware-jamal-khashoggi-murder-nso-hanan-elatr-new-analysis>

<https://www.theguardian.com/news/2021/aug/02/pegasus-spyware-found-on-journalists-phones-french-intelligence-confirms>

Further reading

- <http://en.wikipedia.org/wiki/Stuxnet>
- <http://www.emptywheel.net/2011/10/20/did-duqu-fix-the-bug-that-revealed-stuxnet/>
- <http://en.wikipedia.org/wiki/SCADA>
- <http://www.anti-virus.by/press/viruses/3971.html>
- <http://blogs.technet.com/b/mmpc/archive/2010/07/16/the-stuxnet-sting.aspx>
- <http://eugene.kaspersky.com/2011/11/02/the-man-who-found-stuxnet-sergey-ulasen-in-the-spotlight/>
- <http://krypt3ia.wordpress.com/2011/08/04/plc-controlers-stuxnet-and-kinetic-attacks-blackhat-2011/>
- http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf
- <https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/#slide-1>
- <https://www.wired.com/2012/03/duqu-mystery-language-solved/>
- Merk at flere av kildene på internet er motstridende, spesielt mange avis artikler som hevder å referere til "innside kilder" gir et forvrengt bilde av hvor omfattende Stuxnet angrepet var.
- Informasjon har også dukket opp underveis som kan være i strid med tidligere informasjon, og i strid med hva vi som jobbet med det observerte av oppførsel – er noe misinformasjon? :-)

Øvingsoppgave og arbeidskrav

Arbeidskrav må være godkjent for å ta eksamen i TK2100!

- Selvstudie
- Les om Stuxnet viruset
- Sitt 2-4 studenter sammen og diskuter emnet, legg fokus på potensielle fremtidige scenarioer – og hvor sannsynlig er egentlig skrekkscenarioer innen dette feltet?
- Hvem (i verden) tror du har evnen til å lage slike angrep?
- Veilederne og jeg er tilgjengelig under øvingstimen for å svare på spørsmål, hjelpe dere i research og for å diskutere spennende problemstillinger

ARBEIDSKRAV TK2100 2022:

- Skriv et kort essay (en side eller to) om Stuxnet
 - Hva ble angrepet
 - Hvem stod bak, og hvorfor
 - Lag ditt eget skrekkscenario for fremtiden
- Individuell innlevering. Frist 7. april, detaljer kommer som kunngjøring på Canvas

Tilleggsliteratur om rootkits

- Ikke pensum, men for spesielt interesserte
- Anbefales å ha vært gjennom C programmering som fag eller egenstudie før man begynner på denne
- Den absolutte «bibelen» for å skrive egne rootkits!

