Digitaalinen haavoittuvuus
MATINE
8.5.2018 Tampere

Mika.Rautila@vtt.fi

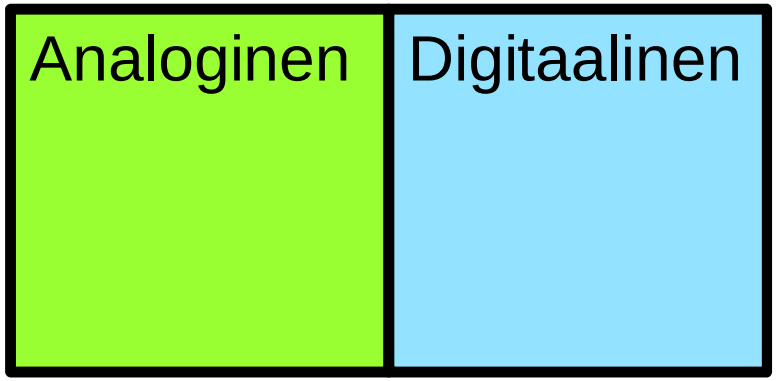
Sisältö

- Mikä on digitaalinen haavoittuvuus
- Millaisista tekijöistä digitaaliset haavoittuvuudet muodostuvat
- Miten haavoittuvuuksia voisi estää

Digitaalinen haavoittuvuus

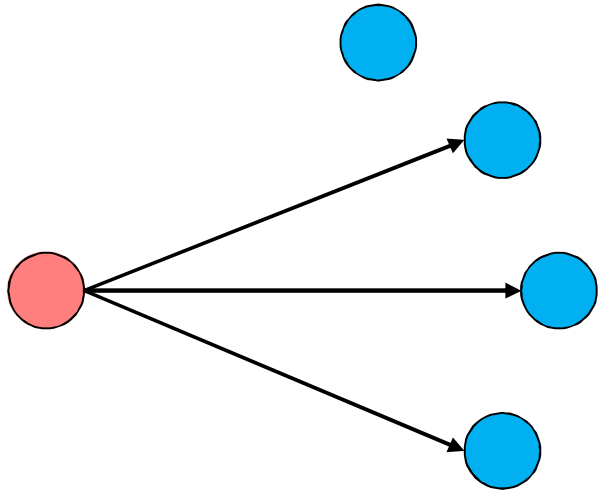
- Järjestelmä voidaan jakaa digitaaliseen ja analogiseen osaan

- Digitaalinen haavoittuvuus on järjestelmän digitaalisessa osassa oleva ominaisuus, joka tekee mahdolliseksi järjestelmän käyttämisen ei-tarkoitetulla (tietoturvapolitiikan vastaisella) tavalla



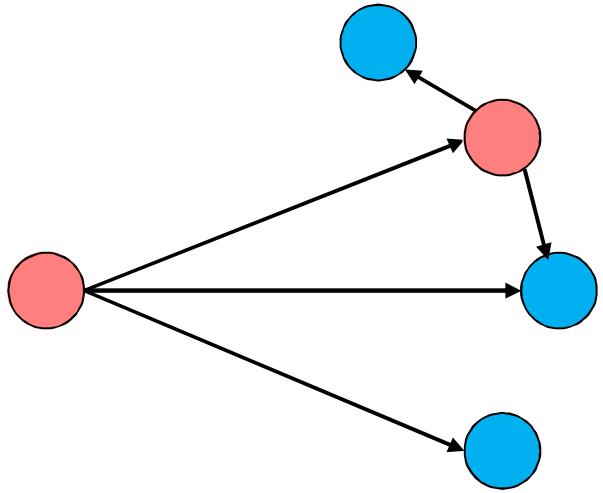
Morris worm

- Tarkoituksena levittäytyä verkossa mahdollisimman moneen koneeseen
- Levittäytyminen käynnistyi lokakuussa 1988 yhdestä MIT:n koneesta



Morris worm

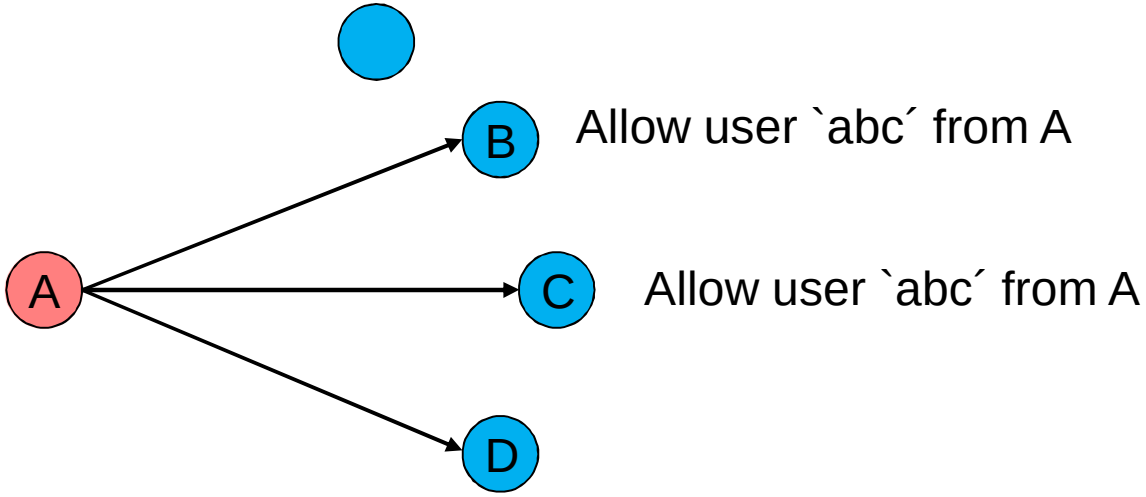
- Tarkoituksena levittäytyä verkossa mahdollisimman moneen koneeseen
- Levittäytyminen käynnistyi lokakuussa 1988 yhdestä MIT:n koneesta



Morris worm

Käytti kolmea eri haavoittuvuutta:

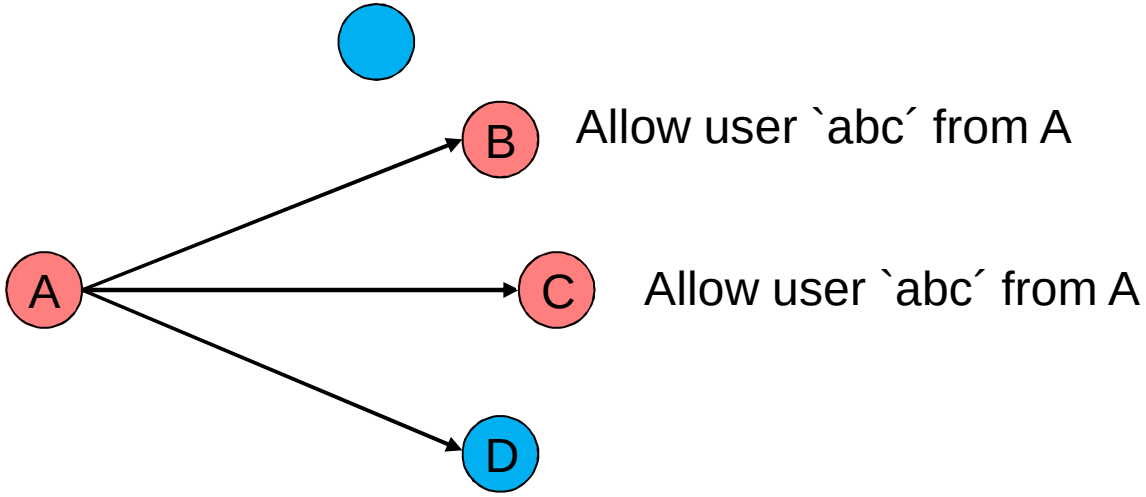
1. Helppokäyttöinen ohjelmien etäsuoritus (rsh)
   - Kaikkien käyttäjien salasanat oli saatavilla salatussa muodossa
   - Helposti arvattavia salasanoja



Morris worm

Käytti kolmea eri haavoittuvuutta:

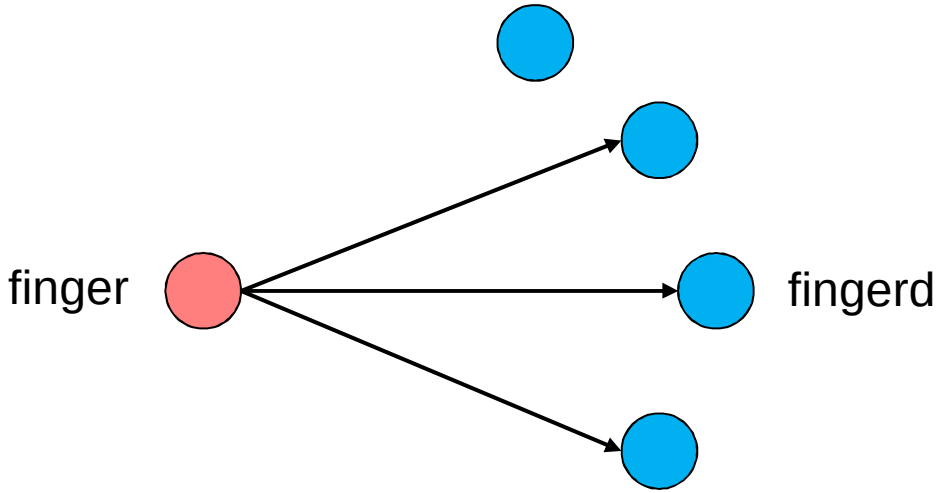
1. Helppokäyttöinen ohjelmien etäsuoritus (rsh)
   - Kaikkien käyttäjien salasanat oli saatavilla salatussa muodossa
   - Helposti arvattavia salasanoja



Morris worm

Käytti kolmea vaihtoehtoista haavoittuvuutta:

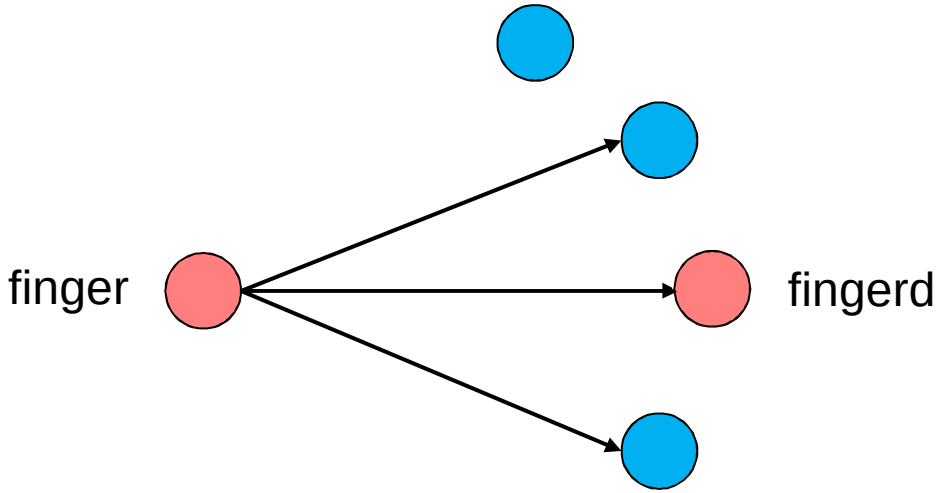
2. Yhden verkkopalvelun toteuttavan ohjelman ohjelmointivirhe (fingerd)
   - Virhe mahdollisti käyttäjän antaman koodin suorittamisen kohdekoneessa



Morris worm

Käytti kolmea vaihtoehtoista haavoittuvuutta:

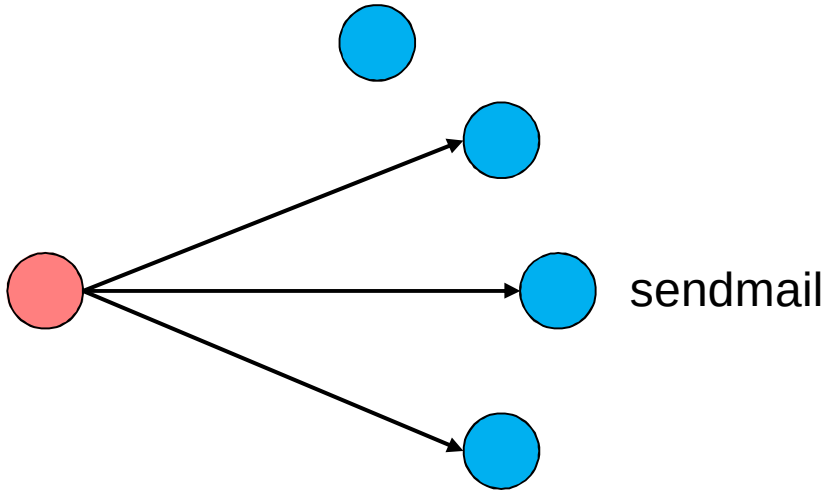
2. Yhden verkkopalvelun toteuttavan ohjelman ohjelmointivirhe (fingerd)
   - Virhe mahdollisti käyttäjän antaman koodin suorittamisen kohdekoneessa



Morris worm

Käytti kolmea vaihtoehtoista haavoittuvuutta:

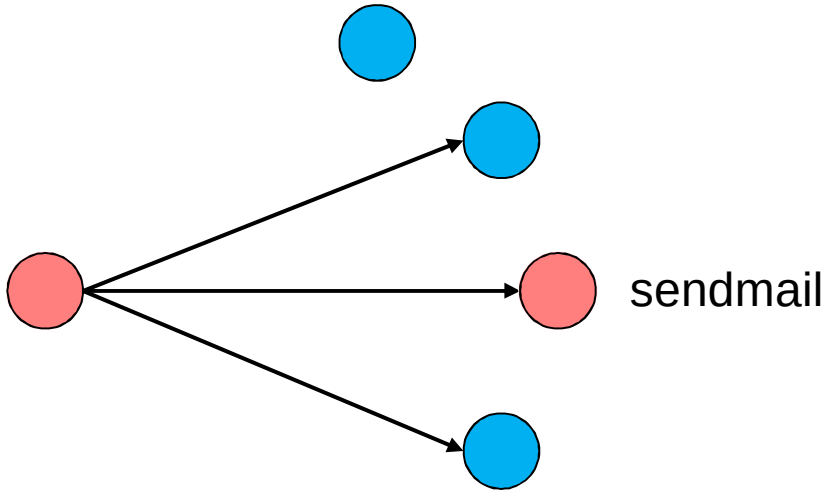
3. Sähköpostiongelmien ratkaisemista helpottava takaportti (sendmail)
   - Ominaisuus oli tarkoitettu käytettäväksi kehitysvaiheessa



Morris worm

Käytti kolmea vaihtoehtoista haavoittuvuutta:

3. Sähköpostiongelmien ratkaisemista helpottava takaportti (sendmail)
   - Ominaisuus oli tarkoitettu käytettäväksi kehitysvaiheessa



Morris worm

Käytti kolmea eri haavoittuvuutta:

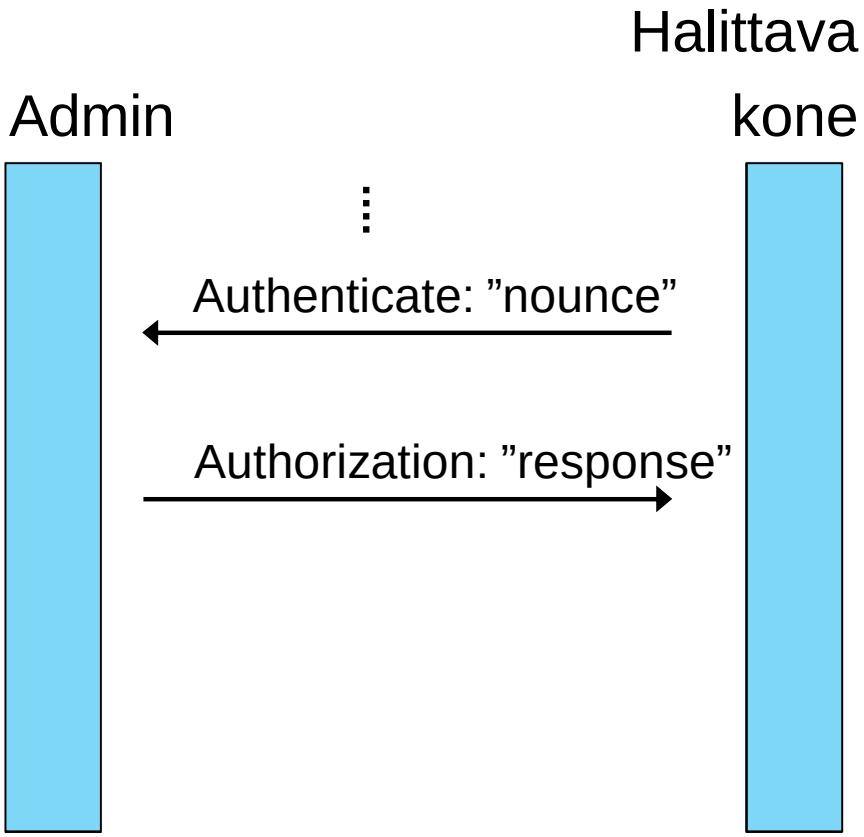
1. Helppokäyttöinen ohjelmien etäsuoritus (rsh)
   - Kaikkien käyttäjien salasanat oli saatavilla salatussa muodossa
   - Helposti arvattavia salasanoja
2. Yhden verkkopalvelun toteuttavan ohjelman ohjelmointivirhe (fingerd)
   - Virhe mahdollisti käyttäjän antaman koodin suorittamisen kohdekoneessa
3. Sähköpostiongelmien ratkaisemista helpottava takaportti (sendmail)
   - Ominaisuus oli tarkoitettu käytettäväksi kehitysvaiheessa

Silent bob

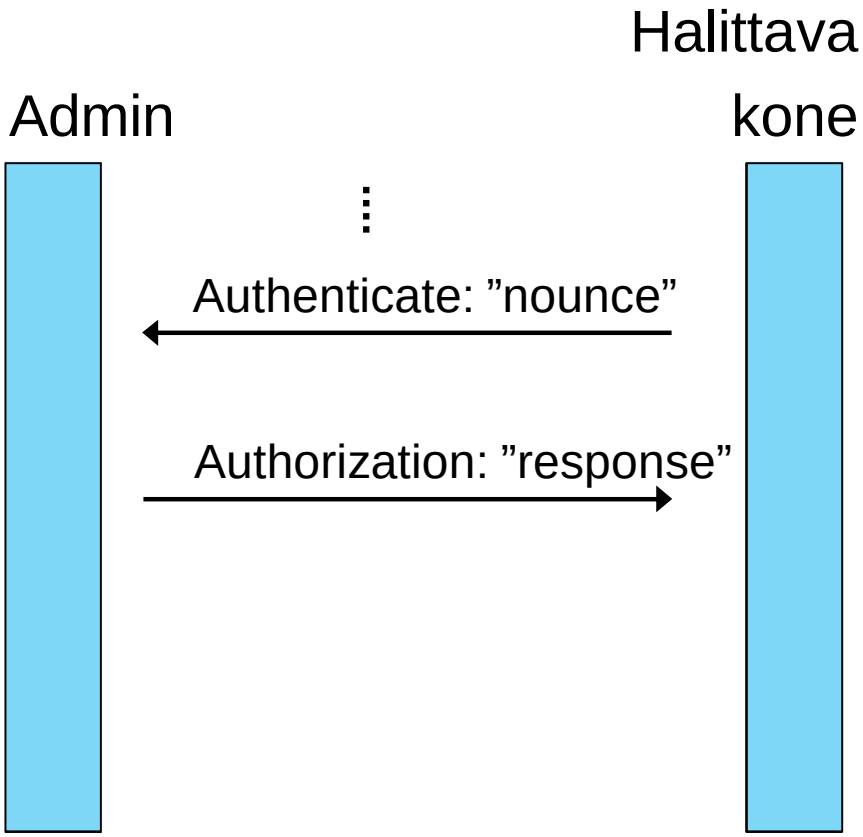
- Haavoittuvuus raportoitu alkuvuodesta 2017
- Intelin PC-arkkitehtuuriin on kehitetty laitteiden etähallinnan mahdollistavat ominaisuudet (Active Management Technology)
- Etähallinta on tehty riippumattomaksi hallittavassa koneessa olevasta käyttöjärjestelmästä
- Hallintaominaisuudet on lähes kaikissa Intelin piireillä valmistetuissa PC-laitteissa
- Etähallinta on toteutettu web-palveluna

- Haavoittuvuus autentikoinnissa web-palveluun

Silent bob

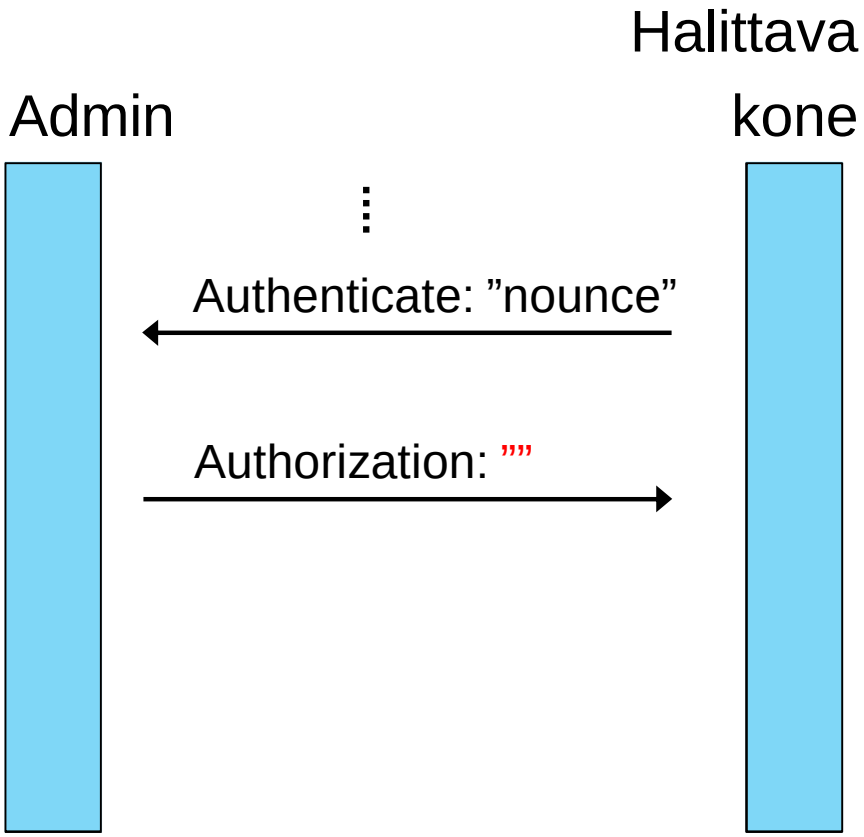

Halittava
Admin
kone
Authenticate: ”nounce”
Authorization: ”response”

Silent bob



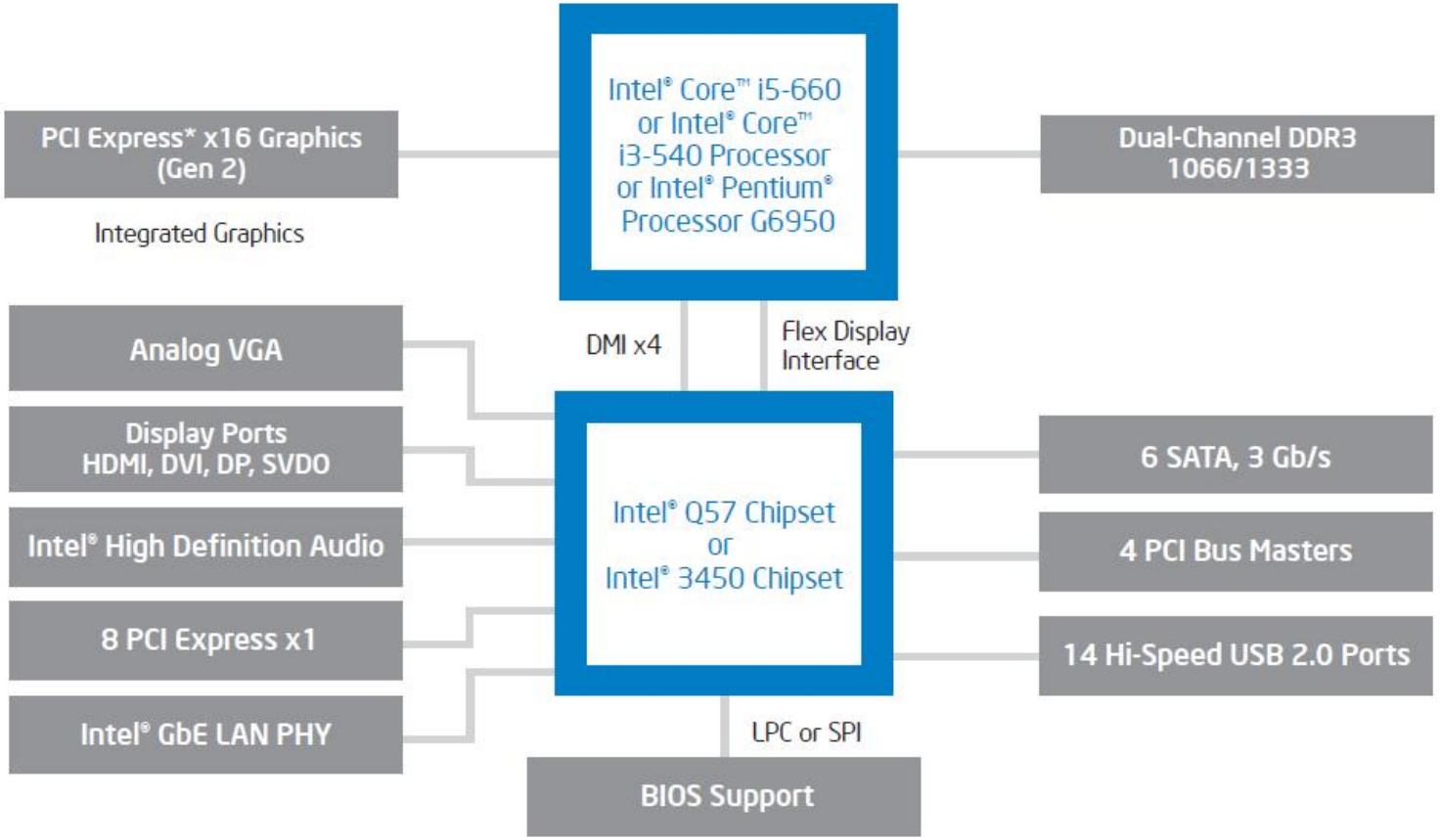

```
if ( strncmp( computed_response,
                user_response,
                response_len ) != 0 )
{
    // Authentication failed.
    // ...
}
```

Silent bob



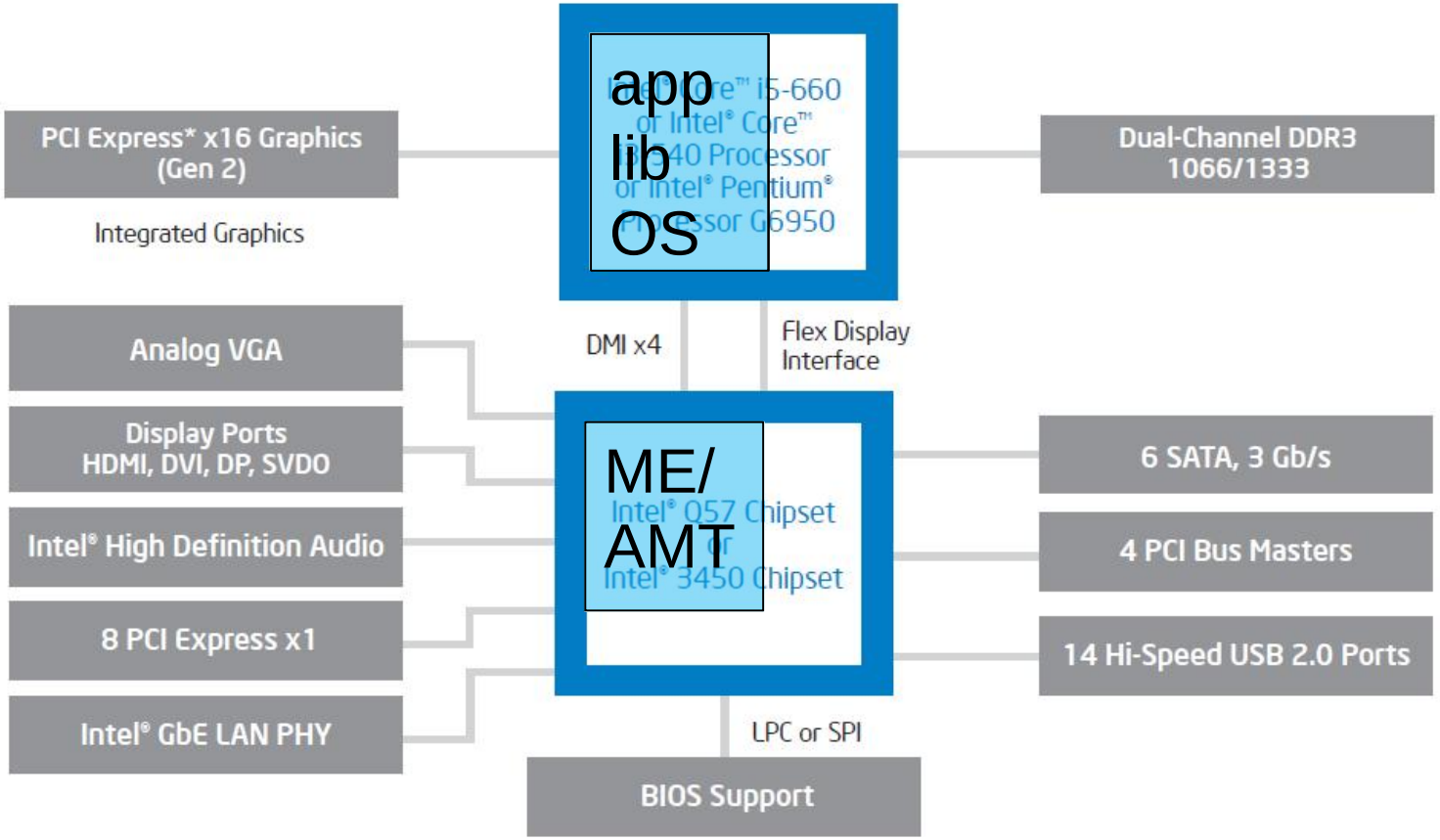

```
⋮
if ( strncmp( computed_response,
                user_response,
                response_len ) != 0 )
{
    // Authentication failed.
    // ...
}
```

Silent bob



Source: Intel product brief -- Intel® Q57 Chipset and Intel® 3450 Chipset Embedded Computing

Silent bob



Source: Intel product brief -- Intel® Q57 Chipset and Intel® 3450 Chipset Embedded Computing

Miten haavoittuvuuksia torjutaan?

Miten haavoittuvuuksia torjutaan?

- Pidetään huoli, ettei ohjelmistoissa ole haavoittuvuuksia
  - Toteutusvaiheen aikainen varmistus
    - Staattinen ja dynaaminen tietoturvan varmistus
  - Miten analysoida vielä tuntemattomien hyökkäystapojen suhteen

Miten haavoittuvuuksia torjutaan?

- Pidetään huoli, ettei ohjelmistoissa ole haavoittuvuuksia
  - Toteutusvaiheen aikainen varmistus
    - Staattinen ja dynaaminen tietoturvan varmistus
  - Miten analysoida vielä tuntemattomien hyökkäystapojen suhteen
- Jos haavoittuvuuksia kuitenkin on, niin pidetään huoli, ettei niitä voi käyttää
  - Defence-in-depth
  - Tietoturvapäivitykset

Miten haavoittuvuuksia torjutaan?

- Pidetään huoli, ettei ohjelmistoissa ole haavoittuvuuksia
  - Toteutusvaiheen aikainen varmistus
    - Staattinen ja dynaaminen tietoturvan varmistus
  - Miten analysoida vielä tuntemattomien hyökkäystapojen suhteen
- Jos haavoittuvuuksia kuitenkin on, niin pidetään huoli, ettei niitä voi käyttää
  - Defence-in-depth
  - Tietoturvapäivitykset
- Pidetään huoli, että käyttäjät käyttävät laitteita turvallisella tavalla

Miten haavoittuvuuksia torjutaan?

- Huono suunnittelu ja konfigurointi
  - Miten analysoida suunnitelmista eri elementtien tietoturvavaikutukset

Miten haavoittuvuuksia torjutaan?

- Huono suunnittelu ja konfigurointi
  - Miten analysoida suunnitelmista eri elementtien tietoturvavaikutukset
- Ohjelman tuotantoversiossa mukana vain kehitysversioon tarkoitettua koodia
  - Miten varmistetaan ettei ole ylimääräistä koodia
  - Miten varmistetaan kaikkien toimittajien osuus
  - Insider-riski
  - Logistiikkariski

Miten haavoittuvuuksia torjutaan?

- Huono suunnittelu ja konfigurointi
  - Miten analysoida suunnitelmista eri elementtien tietoturvavaikutukset
- Ohjelman tuotantoversiossa mukana vain kehitysversioon tarkoitettua koodia
  - Miten varmistetaan ettei ole ylimääräistä koodia
  - Miten varmistetaan kaikkien toimittajien osuus
  - Insider-riski
  - Logistiikkariski
- Ohjelmointivirheet
  - Miten estetään kriittiset ohjelmointivirheet

Miten haavoittuvuuksia torjutaan?

- Huono suunnittelu ja konfigurointi
  - Miten analysoida suunnitelmista eri elementtien tietoturvavaikutukset
- Ohjelman tuotantoversiossa mukana vain kehitysversioon tarkoitettua koodia
  - Miten varmistetaan ettei ole ylimääräistä koodia
  - Miten varmistetaan kaikkien toimittajien osuus
  - Insider-riski
  - Logistiikkariski
- Ohjelmointivirheet
  - Miten estetään kriittiset ohjelmointivirheet
- Hardwaren turvallisuus

Miten haavoittuvuuksia torjutaan?

“You can't trust code that you did not totally create yourself. (Especially code from companies that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code.”

Ken Thompson: Reflections on trusting trust, 1984

Miten haavoittuvuuksia torjutaan?

“You can't trust code that you did not totally create yourself. (Especially code from companies that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code.”

Ken Thompson: Reflections on trusting trust, 1984

“You can’t even trust code that you did totally create yourself”

Eric Allman


TECHNOLOGY FOR BUSINESS