We have to **watch and listen to everything that people are doing** so that we can catch terrorists, drug dealers, pedophiles, and organized criminals. Some of this data is sent unencrypted through the Internet, or sent encrypted to a company that passes the data along to us, but we learn much more when we have **comprehensive direct access to hundreds of millions of disks and screens and microphones and cameras**.

This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Making sure software stays insecure

Daniel J. Bernstein

University of Illinois at Chicago & Technische Universiteit Eindhoven
We have to watch and listen to everything that people are doing so that we can catch terrorists, drug dealers, pedophiles, and organized criminals. Some of this data is sent unencrypted through the Internet, or sent encrypted to a company that passes the data along to us, but we learn much more when we have comprehensive direct access to hundreds of millions of disks and screens and microphones and cameras.

This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Making sure software stays insecure

Daniel J. Bernstein
University of Illinois at Chicago & Technische Universiteit Eindhoven

Some important clarifications:
1. “We” doesn’t include me.
I want secure software.
We have to watch and listen to everything that people are doing so that we can catch terrorists, drug dealers, pedophiles, and organized criminals. Some of this data is sent unencrypted through the Internet, or sent encrypted to a company that passes the data along to us, but we learn much more when we have comprehensive direct access to hundreds of millions of disks and screens and microphones.

This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Some important clarifications:
1. “We” doesn’t include me. I want secure software.
This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Making sure software stays insecure

Daniel J. Bernstein
University of Illinois at Chicago & Technische Universiteit Eindhoven

Some important clarifications:
1. “We” doesn’t include me.
I want secure software.
This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Making sure software stays insecure

Daniel J. Bernstein

University of Illinois at Chicago & Technische Universiteit Eindhoven

Some important clarifications:

1. “We” doesn’t include me.
I want secure software.
This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Making sure software stays insecure

Daniel J. Bernstein
University of Illinois at Chicago & Technische Universiteit Eindhoven

Some important clarifications:

1. “We” doesn’t include me. I want secure software.

2. Their actions violate fundamental human rights.
This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Making sure software stays insecure

Daniel J. Bernstein

University of Illinois at Chicago & Technische Universiteit Eindhoven

Some important clarifications:
1. “We” doesn’t include me. I want secure software.
2. Their actions violate fundamental human rights.
3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.
This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Some important clarifications:
1. “We” doesn’t include me. I want secure software.
2. Their actions violate fundamental human rights.
3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.

Making sure software stays insecure

Daniel J. Bernstein

University of Illinois at Chicago & Technische Universiteit Eindhoven

This talk is actually a thought experiment: how could an attacker manipulate the ecosystem for insecurity?
This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to a wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Some important clarifications:
1. “We” doesn’t include me. I want secure software.
2. Their actions violate fundamental human rights.
3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.

This talk is actually a thought experiment: how could an attacker manipulate the ecosystem for insecurity?

Distract managers, sysadmins, etc. Identify activities that can’t produce secure software but that can nevertheless be marketed as “security.”

Example: virus scanners.

Divert attention, funding, human resources, etc. into “security,” away from actual security.
This talk explains how we’ve successfully manipulated the world’s software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Some important clarifications:
1. “We” doesn’t include me. I want secure software.
2. Their actions violate fundamental human rights.
3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.

This talk is actually a thought experiment: how could an attacker manipulate the ecosystem for insecurity?

Distract managers, sysadmins, etc.
Identify activities that can’t produce secure software but that can nevertheless be marketed as “security”.
Example: virus scanners.
Divert attention, funding, human resources, etc. into “security”, away from actual security.
This talk explains how we've successfully manipulated the world's software ecosystem to ensure our continuing access to this wealth of data. This talk will not cover our efforts against encryption, and will not cover our hardware back doors.

Making sure software stays insecure

Daniel J. Bernstein
University of Illinois at Chicago & Technische Universiteit Eindhoven

Some important clarifications:
1. “We” doesn’t include me. I want secure software.
2. Their actions violate fundamental human rights.
3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.

This talk is actually a thought experiment: how could an attacker manipulate the ecosystem for insecurity?

Distract managers, sysadmins, etc.

Identify activities that can’t produce secure software but that can nevertheless be marketed as “security”.

Example: virus scanners.

Divert attention, funding, human resources, etc. into “security”, away from actual security.
Some important clarifications:

1. “We” doesn’t include me. I want secure software.

2. Their actions violate fundamental human rights.

3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.

This talk is actually a thought experiment: how could an attacker manipulate the ecosystem for insecurity?

Distract managers, sysadmins, etc.

Identify activities that can’t produce secure software but that can nevertheless be marketed as “security”.

Example: virus scanners.

Divert attention, funding, human resources, etc. into “security”, away from actual security.
Some important clarifications:

1. “We” doesn’t include me. I want secure software.

2. Their actions violate fundamental human rights.

3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.

This talk is actually a thought experiment: how could an attacker manipulate the ecosystem for insecurity?

Distract managers, sysadmins, etc.
Identify activities that can’t produce secure software but that can nevertheless be marketed as “security”.

Example: virus scanners.
Divert attention, funding, human resources, etc. into “security”, away from actual security.

People naturally do this.
Attacker investment is magnified. Attack discovery is unlikely.
Important clarifications:

1. "We" doesn’t include me.

I want secure software.

2. Their actions violate fundamental human rights.

I don’t have evidence that deliberately manipulated software ecosystem.

This talk is actually a thought experiment:

Could an attacker manipulate system for insecurity?

Distract managers, sysadmins, etc.

Identify activities that can’t produce secure software but that can nevertheless be marketed as “security”.

Example: virus scanners.

Divert attention, funding, human resources, etc. into “security”, away from actual security.

People naturally do this.

Attacker investment is magnified. Attack discovery is unlikely.

2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . .
Some important clarifications:

1. *We* doesn’t include me.
   
   I want secure software.

2. Their actions violate fundamental human rights.

3. I don’t have evidence that they’ve deliberately manipulated the software ecosystem.

This talk is actually a thought experiment: how could an attacker manipulate the ecosystem for insecurity?

- Distract managers, sysadmins, etc.
- Identify activities that *can’t* produce secure software but that can nevertheless be marketed as “security”.
- Example: virus scanners.
- Divert attention, funding, human resources, etc. into “security”, away from actual security.

**People naturally do this.**

Attacker investment is magnified. Attack discovery is unlikely.

2014 NIST “Framework for improving critical cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety at risk. . . .
Distract managers, sysadmins, etc.
Identify activities that can’t produce secure software but that can nevertheless be marketed as “security”.
Example: virus scanners.
Divert attention, funding, human resources, etc. into “security”, away from actual security.

People naturally do this.
Attacker investment is magnified.
Attack discovery is unlikely.

2014 NIST “Framework for improving critical infrastructure cybersecurity”:
“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . .
Distract managers, sysadmins, etc.
Identify activities that can’t produce secure software but that can nevertheless be marketed as “security”.
Example: virus scanners.
Divert attention, funding, human resources, etc. into “security”, away from actual security.

People naturally do this.
Attacker investment is magnified. Attack discovery is unlikely.

2014 NIST “Framework for improving critical infrastructure cybersecurity”:
“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . .
Distract managers, sysadmins, etc.

Identify activities that *can’t* produce secure software but that can nevertheless be marketed as “security”.

Example: virus scanners.

Divert attention, funding, human resources, etc. into “security”, away from actual security.

**People naturally do this.**

Attacker investment is magnified. Attack discovery is unlikely.

---

2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”
Distract managers, sysadmins, etc. Identify activities that cannot produce secure software but that can nevertheless be marketed as “security”. Example: virus scanners. Divert attention, funding, human resources, etc. into “security”, away from actual security. People naturally do this. Attacker investment is magnified. Attack discovery is unlikely.

2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to achieve cybersecurity goals in a cost-effective, prioritized manner.”
Distract managers, sysadmins, etc.

Identify activities that can't produce secure software but that can nevertheless be marketed as "security".

Example: virus scanners.

Divert attention, funding, human resources, etc. into "security", away from actual security.

People naturally do this.

Attacker investment is magnified.

Attack discovery is unlikely.

2014 NIST "Framework for improving critical infrastructure cybersecurity":

"Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation's security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization's risk management processes."

"This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner."
Distract managers, sysadmins, etc. Identify activities that can’t produce secure software but that can nevertheless be marketed as “security.” Example: virus scanners. Divert attention, funding, human resources, etc. into “security,” away from actual security. People naturally do this. Attacker investment is magnified. Attack discovery is unlikely.

2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”
2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”
2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  e. g. inventory your PCs.
2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  e.g. inventory your PCs.
- “Protect.”
  e.g. inventory your humans.
2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  e.g. inventory your PCs.
- “Protect.”
  e.g. inventory your humans.
- “Detect.”
  e.g. install an IDS.
2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  e.g. inventory your PCs.
- “Protect.”
  e.g. inventory your humans.
- “Detect.”
  e.g. install an IDS.
- “Respond.”
  e.g. coordinate with CERT.
2014 NIST “Framework for improving critical infrastructure cybersecurity”:

“Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, and public safety and health at risk. . . . The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.”

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  e.g. inventory your PCs.
- “Protect.”
  e.g. inventory your humans.
- “Detect.”
  e.g. install an IDS.
- “Respond.”
  e.g. coordinate with CERT.
- “Recover.”
  e.g. “Reputation is repaired.”
“Framework for improving critical infrastructure cybersecurity”:

Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation’s security, economy, public safety and health at risk.

The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization’s risk management processes.

“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  - e.g. inventory your PCs.

- “Protect.”
  - e.g. inventory your humans.

- “Detect.”
  - e.g. install an IDS.

- “Respond.”
  - e.g. coordinate with CERT.

- “Recover.”
  - e.g. “Reputation is repaired.”

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  - e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  - e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  - e.g. review your audit logs.
Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation's security, economy, and public safety and health at risk. The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization's risk management processes. This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.

Categories inside "Protect":
- Access Control
- Awareness and Training
- Data Security
- Information Protection Processes and Procedures
- Maintenance
- Protective Technology

"Identify." e.g. inventory your humans.
"Protect." e.g. inventory your PCs.
"Detect." e.g. install an IDS.
"Respond." e.g. coordinate with CERT.
"Recover." e.g. "Reputation is repaired."
2014 NIST "Framework for improving critical infrastructure cybersecurity":
"Cybersecurity threats exploit the increased complexity and connectivity of critical infrastructure systems, placing the Nation's security, economy, and public safety and health at risk.

The Framework focuses on using business drivers to guide cybersecurity activities and considering cybersecurity risks as part of the organization's risk management processes."

This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.

- "Identify."
  e.g. inventory your PCs.
- "Protect."
  e.g. inventory your humans.
- "Detect."
  e.g. install an IDS.
- "Respond."
  e.g. coordinate with CERT.
- "Recover."
  e.g. “Reputation is repaired.”

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  e.g. review your audit logs.
“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  - e.g. inventory your PCs.
- “Protect.”
  - e.g. inventory your humans.
- “Detect.”
  - e.g. install an IDS.
- “Respond.”
  - e.g. coordinate with CERT.
- “Recover.”
  - e.g. “Reputation is repaired.”

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  - e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  - e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  - e.g. review your audit logs.
“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  e.g. inventory your PCs.
- “Protect.”
  e.g. inventory your humans.
- “Detect.”
  e.g. install an IDS.
- “Respond.”
  e.g. coordinate with CERT.
- “Recover.”
  e.g. “Reputation is repaired.”

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  e.g. review your audit logs.

Subcategories in Framework: 98.

... promoting secure software: 0.
“This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.”

- “Identify.”
  e.g. inventory your PCs.

- “Protect.”
  e.g. inventory your humans.

- “Detect.”
  e.g. install an IDS.

- “Respond.”
  e.g. coordinate with CERT.

- “Recover.”
  e.g. “Reputation is repaired.”

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  e.g. review your audit logs.

Subcategories in Framework: 98.
... promoting secure software: 0.

This is how the money is spent.
This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.

**Identify.**
e.g. inventory your PCs.

**Protect.**
e.g. inventory your humans.

**Detect.**
e.g. install an IDS.

**Respond.**
e.g. coordinate with CERT.

**Recover.**
e.g. "Reputation is repaired."

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
e.g. inventory your data.
- “Information Protection Processes and Procedures”.
e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
e.g. review your audit logs.

Subcategories in Framework: 98.
... promoting secure software: 0.

This is how the money is spent.

Distract users
e.g. “Download only trusted applications from reputable sources or marketplaces.”
e.g. “Be suspicious of unknown links or requests sent through email or text.”
e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”
e.g. “Ideally, you will have separate computers for work and personal.”
This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.

- **Identify.** e.g. inventory your PCs.
- **Protect.** e.g. inventory your humans.
- **Detect.** e.g. install an IDS.
- **Respond.** e.g. coordinate with CERT.
- **Recover.** e.g. “Reputation is repaired.”

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  e.g. review your audit logs.

Subcategories in Framework: 98.
... promoting secure software: 0.

This is how the money is spent.

Distract users
 e.g. “Download only trusted applications from reputable sources or marketplaces.”
 e.g. “Be suspicious of unknown links or requests sent through email or text message.”
 e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”
 e.g. “Ideally, you will have separate computers for work and personal use.”
This risk-based approach enables an organization to gauge resource estimates (e.g., staffing, funding) to achieve cybersecurity goals in a cost-effective, prioritized manner.

Identify.
- e.g. inventory your PCs.

Protect.
- e.g. inventory your humans.

Detect.
- e.g. install an IDS.

Respond.
- e.g. coordinate with CERT.

Recover.
- e.g. "Reputation is repaired."

Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  - e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  - e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  - e.g. review your audit logs.

Subcategories in Framework: 98.
... promoting secure software: 0.

This is how the money is spent.

Distract users
- e.g. “Download only trusted applications from reputable sources or marketplaces.”
- e.g. “Be suspicious of unknown links or requests sent through email or text message.”
- e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”
- e.g. “Ideally, you will have separate computers for work and personal use.”
Categories inside “Protect”:
- “Access Control”.
- “Awareness and Training”.
- “Data Security”.
  e.g. inventory your data.
- “Information Protection Processes and Procedures”.
  e.g. inventory your OS versions.
- “Maintenance”.
- “Protective Technology”.
  e.g. review your audit logs.

Subcategories in Framework: 98.

... promoting secure software: 0.

This is how the money is spent.

Distract users

e.g. “Download only trusted applications from reputable sources or marketplaces.”

e.g. “Be suspicious of unknown links or requests sent through email or text message.”

e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”

e.g. “Ideally, you will have separate computers for work and personal use.”
Categories inside “Protect”:
- Access Control
- Awareness and Training
- Data Security
- Information Protection Processes and Procedures
- Maintenance
- Protective Technology

e.g. inventory your data.
e.g. inventory your OS versions.
e.g. review your audit logs.

Subcategories in Framework: 98.

- Distract users
  - e.g. “Download only trusted applications from reputable sources or marketplaces.”
  - e.g. “Be suspicious of unknown links or requests sent through email or text message.”
  - e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”
  - e.g. “Ideally, you will have separate computers for work and personal use.”

- Distract programmers
  - Example: automatic low-latency software “security” updates.
Categories inside “Protect”:

“Access Control”.
“Awareness and Training”.
“Data Security”.

Subcategories in Framework: 98.

Distract users

e.g. “Download only trusted applications from reputable sources or marketplaces.”

e.g. “Be suspicious of unknown links or requests sent through email or text message.”

e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”

e.g. “Ideally, you will have separate computers for work and personal use.”

Distract programmers

Example: automatic low-latency software “security” updates.
Distract users

e.g. “Download only trusted applications from reputable sources or marketplaces.”

e.g. “Be suspicious of unknown links or requests sent through email or text message.”

e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”

e.g. “Ideally, you will have separate computers for work and personal use.”

Distract programmers

Example: automatic low-latency software “security” updates.
Distract users

e.g. “Download only trusted applications from reputable sources or marketplaces.”

e.g. “Be suspicious of unknown links or requests sent through email or text message.”

e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”

e.g. “Ideally, you will have separate computers for work and personal use.”

Distract programmers

Example: automatic low-latency software “security” updates.
Distract users

e.g. “Download only trusted applications from reputable sources or marketplaces.”

e.g. “Be suspicious of unknown links or requests sent through email or text message.”

e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”

e.g. “Ideally, you will have separate computers for work and personal use.”

Distract programmers

Example: automatic low-latency software “security” updates.

Marketing: “security” is defined by public security holes.
Known hole in Product 2014.06? Update now to Product 2014.07!
Distract users

- e.g. “Download only trusted applications from reputable sources or marketplaces.”
- e.g. “Be suspicious of unknown links or requests sent through email or text message.”
- e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”
- e.g. “Ideally, you will have separate computers for work and personal use.”

Distract programmers

- Example: automatic low-latency software “security” updates.
- Marketing: “security” is defined by *public security holes*. Known hole in Product 2014.06? Update now to Product 2014.07!
- To help the marketing, publicize actual attacks that exploit public security holes.
Distract users

e.g. “Download only trusted applications from reputable sources or marketplaces.”

e.g. “Be suspicious of unknown links or requests sent through email or text message.”

e.g. “Immediately report any suspect data or security breaches to your supervisor and/or authorities.”

e.g. “Ideally, you will have separate computers for work and personal use.”

Distract programmers

Example: automatic low-latency software “security” updates.

Marketing: “security” is defined by public security holes.
Known hole in Product 2014.06? Update now to Product 2014.07!

To help the marketing, publicize actual attacks that exploit public security holes.

Reality: Product 2014.07 also has security holes that attackers are exploiting.
Distract users

download only trusted applications from reputable sources or marketplaces.

be suspicious of unknown requests sent through text message.

immediately report any data or security breaches to your supervisor and/or authorities.

really, you will have separate computers for work and use.

Distract programmers

Example: automatic low-latency software “security” updates.

Marketing: “security” is defined by public security holes.

Known hole in Product 2014.06? Update now to Product 2014.07!

To help the marketing, publicize actual attacks that exploit public security holes.

Reality: Product 2014.07 also has security holes that attackers are exploiting.

Distract researchers

Example:

When researcher finds attack showing that a system is insecure, create a competition for the amount of damage.

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”
Distract users

- e.g. "Download only trusted applications from reputable sources or marketplaces."
- e.g. "Be suspicious of unknown links or requests sent through email or text message."
- e.g. "Immediately report any suspect data or security breaches to your supervisor and/or authorities."
- e.g. "Ideally, you will have separate computers for work and personal use."

Distract programmers

Example: automatic low-latency software “security” updates.

Marketing: “security” is defined by *public security holes*.
Known hole in Product 2014.06? Update now to Product 2014.07!

To help the marketing, publicize actual attacks that exploit public security holes.

Reality: Product 2014.07 also has security holes that attackers are exploiting.

Distract researchers

Example:

When researcher finds attack showing that a system is insecure, create a competition for the amount of damage that

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”
Distract programmers

Example: automatic low-latency software “security” updates.

Marketing: “security” is defined by *public security holes*.

Known hole in Product 2014.06? Update now to Product 2014.07!

To help the marketing, publicize actual attacks that exploit public security holes.

Reality: Product 2014.07 also has security holes that attackers are exploiting.

Distract researchers

Example:

When researcher finds attack showing that a system is insecure, create a competition for the *amount of damage*.

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”
Distract programmers
Example: automatic low-latency software “security” updates.
Marketing: “security” is defined by *public security holes*.
Known hole in Product 2014.06? Update now to Product 2014.07!
To help the marketing, publicize actual attacks that exploit public security holes.
Reality: Product 2014.07 also has security holes that attackers are exploiting.

Distract researchers
Example:
When researcher finds attack showing that a system is insecure, create a competition for *the amount of damage*.
“You corrupted only one file?”
“How many users are affected?”
“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”
Distract programmers

Example: automatic low-latency software “security” updates.

Marketing: “security” is defined by *public security holes*.
Known hole in Product 2014.06?
Update now to Product 2014.07!

To help the marketing, publicize actual attacks that exploit public security holes.

Reality: Product 2014.07 also has security holes that attackers are exploiting.

Distract researchers

Example:
When researcher finds attack showing that a system is insecure, create a competition for *the amount of damage*.

“You corrupted only one file?”

“How many users are affected?”

“How do you really expect an attacker to use 100 CPU cores for a month just to break this system?”

⇒ More attack papers!
Distract programmers

Example: automatic low-latency "security" updates.

Marketing: "security" is defined by public security holes.

Known hole in Product 2014.06? Update now to Product 2014.07!

To help the marketing, publicize actual attacks that exploit public security holes.

Reality: Product 2014.07 also has security holes that attackers are exploiting.

---

Distract researchers

Example:
When researcher finds attack showing that a system is insecure, create a competition for the amount of damage.

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”

⇒ More attack papers!

---

Discourage security

Tell programmers that “100% security is impossible” so they shouldn’t even try.
Distract programmers

Example: automatic low-latency software “security” updates.

Marketing: “security” is defined by public security holes.

Known hole in Product 2014.06?

Update now to Product 2014.07!

To help the marketing, publicize actual attacks that exploit public security holes.

Reality: Product 2014.07 also has security holes that attackers are exploiting.

Distract researchers

Example:

When researcher finds attack showing that a system is insecure, create a competition for the amount of damage.

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”

⇒ More attack papers!

Discourage security

Tell programmers that “100% security is impossible” so they shouldn’t even try.
Distract researchers

Example:
When researcher finds attack showing that a system is insecure, create a competition for the amount of damage.

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”

⇒ More attack papers!

Discourage security

Tell programmers that “100% security is impossible” so they shouldn’t even try.
Distract researchers

Example:
When researcher finds attack showing that a system is insecure, create a competition for the amount of damage.

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”

⇒ More attack papers!

Discourage security

Tell programmers that “100% security is impossible” so they shouldn’t even try.
Distract researchers

Example:
When researcher finds attack showing that a system is insecure, create a competition for the amount of damage.

“You corrupted only one file?”

“How many users are affected?”

“How do you really expect an attacker to use 100 CPU cores for a month just to break this system?”

⇒ More attack papers!

Discourage security

Tell programmers that “100% security is impossible” so they shouldn’t even try.

Tell programmers that “defining security is impossible” so it can’t be implemented.
Distract researchers

Example:
When researcher finds attack showing that a system is insecure, create a competition for *the amount of damage*.

“*You corrupted only one file?*”

“How many users are affected?”

“*Do you really expect an attacker to use 100 CPU cores for a month just to break this system?*”

⇒ More attack papers!

Discourage security

Tell programmers that “*100% security is impossible*” so they shouldn’t even try.

Tell programmers that “*defining security is impossible*” so it can’t be implemented.

Hide/dismiss/mismeasure security metric #1.
Distract researchers

Example:
When researcher finds attack showing that a system is insecure, create a competition for the amount of damage.

“You corrupted only one file?”

“How many users are affected?”

“Do you really expect an attacker to use 100 CPU cores for a month just to break this system?”

⇒ More attack papers!

Discourage security

Tell programmers that “100% security is impossible” so they shouldn’t even try.

Tell programmers that “defining security is impossible” so it can’t be implemented.

Hide/dismiss/mismeasure security metric #1.

Prioritize compatibility, “standards”, speed, etc. e.g.:
“An HTTP server in the kernel is critical for performance.”
Discourage security

Tell programmers that
“100% security is impossible”
so they shouldn’t even try.

Tell programmers that
“defining security is impossible”
so it can’t be implemented.

Hide/dismiss/mismeasure security metric #1.

Prioritize compatibility,
“standards”, speed, etc. e.g.:
“An HTTP server in the kernel
is critical for performance.”

What is security?

Integrity policy #1:
Whenever the computer
shows me a file,
it also tells me
the source of the file.

E.g. If Eve creates a file
and convinces the computer
to show me the file
as having source Frank
then this policy is violated.

I have a few other
security policies,
but this is my top priority.
Discourage security

Tell programmers that “100% security is impossible” so they shouldn’t even try.

Tell programmers that “defining security is impossible” so it can’t be implemented.

Hide/dismiss/mismeasure security metric #1.

Prioritize compatibility, “standards”, speed, etc. e.g.:
“An HTTP server in the kernel is critical for performance.”

What is security?

Integrity policy #1:
Whenever the computer shows me a file, it also tells me the source of the file.

e.g. If Eve creates a file and convinces the computer to show me the file as having source Frank then this policy is violated.

I have a few other security policies, but this is my top priority.
Discourage security
Tell programmers that
“100% security is impossible”
so they shouldn’t even try.

Tell programmers that
“defining security is impossible”
so it can’t be implemented.

Hide/dismiss/mismeasure
security metric #1.

Prioritize compatibility,
“standards”, speed, etc. e.g.:
“An HTTP server in the kernel
is critical for performance.”

What is security?
Integrity policy #1:
Whenever the computer
shows me a file,
it also tells me
the source of the file.

e.g. If Eve creates a file
and convinces the computer
to show me the file
as having source Frank
then this policy is violated.

I have a few other
security policies,
but this is my top priority.
Discourage security

Tell programmers that "100% security is impossible" so they shouldn’t even try.

Tell programmers that "defining security is impossible" so it can’t be implemented.

Hide/dismiss/mismeasure security metric #1.

Prioritize compatibility, "standards", speed, etc. e.g.:
"An HTTP server in the kernel is critical for performance."

What is security?

Integrity policy #1:
Whenever the computer shows me a file, it also tells me the source of the file.

e.g. If Eve creates a file and convinces the computer to show me the file as having source Frank then this policy is violated.

I have a few other security policies, but this is my top priority.
Discourage security
Tell programmers that “security is impossible” shouldn’t even try.
Tell programmers that “defining security is impossible” can’t be implemented.
Hide/dismiss/mismeasure security metric #1.
Prioritize compatibility, “standards”, speed, etc. e.g.: “An HTTP server in the kernel is critical for performance.”

What is security?

Integrity policy #1:
Whenever the computer shows me a file, it also tells me the source of the file.
e.g. If Eve creates a file and convinces the computer to show me the file as having source Frank then this policy is violated.

I have a few other security policies, but this is my top priority.

The trusted computing base
1987: My first UNIX experience.
Low-cost terminals access multi-user Ultrix computer.

Picture credit: terminals.classiccmp.org/wiki/index.php/DEC_VT102
What is security?

Integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.

e.g. If Eve creates a file and convinces the computer to show me the file as having source Frank then this policy is violated.

I have a few other security policies, but this is my top priority.
What is security?

Integrity policy #1:
Whenever the computer shows me a file, it also tells me the source of the file.

e.g. If Eve creates a file and convinces the computer to show me the file as having source Frank then this policy is violated.

I have a few other security policies, but this is my top priority.

The trusted computing base

1987: My first UNIX experience.
Low-cost terminals access multi-user Ultrix computer.

Picture credit:
terminals.classiccmp.org/wiki/index.php/DEC_VT102
What is security?

Integrity policy #1:
Whenever the computer shows me a file, it also tells me the source of the file.

e.g. If Eve creates a file and convinces the computer to show me the file as having source Frank then this policy is violated.

I have a few other security policies, but this is my top priority.

The trusted computing base
1987: My first UNIX experience. Low-cost terminals access multi-user Ultrix computer.

Picture credit: terminals.classiccmp.org/wiki/index.php/DEC_VT102
What is security?

Integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.

Eve creates a file and convinces the computer to show me the file having source Frank. This policy is violated.

I have a few other security policies, but this is my top priority.

The trusted computing base

1987: My first UNIX experience. Low-cost terminals access multi-user Ultrix computer.

I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan. (Of course, sysadmin can.)
What is security?

Integrity policy #1:
Whenever the computer shows me a file, it also tells me the source of the file.

E.g. If Eve creates a file and convinces the computer to show me the file as having source Frank then this policy is violated.

I have a few other security policies, but this is my top priority.

The trusted computing base
1987: My first UNIX experience.
Low-cost terminals access multi-user Ultrix computer.

I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan.

(Of course, sysadmin can.)
The trusted computing base
1987: My first UNIX experience. Low-cost terminals access multi-user Ultrix computer.

I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan. (Of course, sysadmin can.)

Picture credit:
terminals.classiccmp.org/wiki/index.php/DEC_VT102
The trusted computing base

1987: My first UNIX experience.
Low-cost terminals access multi-user Ultrix computer.

I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan.
(Of course, sysadmin can.)
I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan.
(Of course, sysadmin can.)
I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan. (Of course, sysadmin can.)

How is this implemented?

OS kernel allocates disk space:

<table>
<thead>
<tr>
<th></th>
<th>system files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my files</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s files</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s files</td>
</tr>
</tbody>
</table>

OS kernel allocates RAM:

<table>
<thead>
<tr>
<th></th>
<th>kernel memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my processes</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s processes</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s processes</td>
</tr>
</tbody>
</table>
I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan.
(Of course, sysadmin can.)

How is this implemented?
OS kernel allocates disk space:

<table>
<thead>
<tr>
<th></th>
<th>system files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my files</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s files</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s files</td>
</tr>
</tbody>
</table>

OS kernel allocates RAM:

<table>
<thead>
<tr>
<th></th>
<th>kernel memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my processes</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s processes</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s processes</td>
</tr>
</tbody>
</table>
I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan. (Of course, sysadmin can.)

How is this implemented?

OS kernel allocates disk space:

<table>
<thead>
<tr>
<th></th>
<th>system files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my files</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s files</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s files</td>
</tr>
</tbody>
</table>

OS kernel allocates RAM:

<table>
<thead>
<tr>
<th></th>
<th>kernel memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my processes</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s processes</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s processes</td>
</tr>
</tbody>
</table>
I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan.
(Of course, sysadmin can.)

How is this implemented?

OS kernel allocates disk space:

<table>
<thead>
<tr>
<th></th>
<th>system files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my files</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s files</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s files</td>
</tr>
</tbody>
</table>

OS kernel allocates RAM:

<table>
<thead>
<tr>
<th></th>
<th>kernel memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my processes</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s processes</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s processes</td>
</tr>
</tbody>
</table>

CPU hardware enforces memory protection: a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules. When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.
I log in to the Ultrix computer, store files labeled Dan, start processes labeled Dan.

Eve logs in, stores files labeled Eve, starts processes labeled Eve.

Frank logs in, stores files labeled Frank, starts processes labeled Frank.

Eve and Frank cannot store files labeled Dan, start processes labeled Dan.
(Of course, sysadmin can.)

How is this implemented?

OS kernel allocates disk space:

<table>
<thead>
<tr>
<th></th>
<th>system files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my files</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s files</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s files</td>
</tr>
</tbody>
</table>

OS kernel allocates RAM:

<table>
<thead>
<tr>
<th></th>
<th>kernel memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my processes</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s processes</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s processes</td>
</tr>
</tbody>
</table>

CPU hardware enforces memory protection:
a user process cannot read or write files or RAM in other processes without permission.

Kernel enforces various rules.

When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.

min can.)
How is this implemented?

OS kernel allocates disk space:

<table>
<thead>
<tr>
<th></th>
<th>system files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my files</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve's files</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank's files</td>
</tr>
</tbody>
</table>

OS kernel allocates RAM:

<table>
<thead>
<tr>
<th></th>
<th>kernel memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my processes</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve's processes</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank's processes</td>
</tr>
</tbody>
</table>

CPU hardware enforces **memory protection**: a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules:

When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.
How is this implemented?

OS kernel allocates disk space:

<table>
<thead>
<tr>
<th></th>
<th>system files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my files</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s files</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s files</td>
</tr>
</tbody>
</table>

OS kernel allocates RAM:

<table>
<thead>
<tr>
<th></th>
<th>kernel memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan</td>
<td>my processes</td>
</tr>
<tr>
<td>Eve</td>
<td>Eve’s processes</td>
</tr>
<tr>
<td>Frank</td>
<td>Frank’s processes</td>
</tr>
</tbody>
</table>

CPU hardware enforces memory protection:
a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules.

When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.
How is this implemented?

OS kernel allocates disk space:
- system files
- my files
- Eve’s files
- Frank’s files

OS kernel allocates RAM:
- kernel memory
- my processes
- Eve’s processes
- Frank’s processes

CPU hardware enforces memory protection: a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules.

When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
How is this implemented?

OS kernel allocates disk space:
- system files
- Dan's files
- Eve's files
- Frank's files

OS kernel allocates RAM:
- kernel memory
- Dan's processes
- Eve's processes
- Frank's processes

CPU hardware enforces memory protection:
- a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules.

When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works.

How do we verify that Eve can't write Dan's files?

1. Check the code that enforces these rules.
CPU hardware enforces memory protection: a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules. When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
CPU hardware enforces memory protection:
a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules.

When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
CPU hardware enforces **memory protection**: a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules.

When a process creates another process or a file, kernel copies uid.

Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.

2. Check the code that allocates disk space, RAM; and user-authentication code.
CPU hardware enforces memory protection: a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules. When a process creates another process or a file, kernel copies uid. Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
2. Check the code that allocates disk space, RAM; and user-authentication code.
3. Check all other kernel code. Bugs anywhere in kernel can override these rules. Memory protection doesn’t apply; language (C) doesn’t compensate.
CPU hardware enforces memory protection: a user process cannot read or write files or RAM in other processes without permission from kernel.

Kernel enforces various rules. When a process creates another process or a file, kernel copies uid. Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
2. Check the code that allocates disk space, RAM; and user-authentication code.
3. Check all other kernel code. Bugs anywhere in kernel can override these rules.

Memory protection doesn’t apply; language (C) doesn’t compensate.

The code we have to check is the trusted computing base. Security metric #1: TCB size.

Eve can’t write Dan’s files unless there's a TCB bug.

Eve’s actions are irrelevant. Other software is irrelevant.

Millions of lines of code that we don’t have to check.

Do we need an audit log? No.
Keep computers separate? No.
Limit software Eve can run? No.
CPU hardware enforces memory protection: a user process cannot read or write files or RAM in other processes without permission from kernel. Kernel enforces various rules. When a process creates another process or a file, kernel copies uid. Process is allowed to read or write any file with the same uid, but not with different uid.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
2. Check the code that allocates disk space, RAM; and user-authentication code.
3. Check all other kernel code.

Bugs anywhere in kernel can override these rules. Memory protection doesn’t apply; language (C) doesn’t compensate.

The code we have to check is the trusted computing base. Security metric #1: TCB size.

Eve can’t write Dan’s files unless there’s a TCB bug.

Eve’s actions: irrelevant. Other software: irrelevant. Millions of lines of code that we don’t have to check.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
2. Check the code that allocates disk space, RAM; and user-authentication code.
3. Check all other kernel code. Bugs anywhere in kernel can override these rules.

Memory protection doesn’t apply; language (C) doesn’t compensate.

The code we have to check is the trusted computing base. Security metric #1: TCB size.

Eve can’t write Dan’s files unless there’s a TCB bug.

Eve’s actions: irrelevant. Other software: irrelevant. Millions of lines of code that we don’t have to check.

Assume the hardware works. How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
2. Check the code that allocates disk space, RAM; and user-authentication code.
3. Check all other kernel code.

Bugs anywhere in kernel can override these rules. Memory protection doesn’t apply; language (C) doesn’t compensate.

The code we have to check is the trusted computing base. Security metric #1: TCB size.

Eve can’t write Dan’s files unless there’s a TCB bug.

Eve’s actions: irrelevant. Other software: irrelevant. Millions of lines of code that we don’t have to check.

Assume the hardware works.

How do we verify that Eve can’t write Dan’s files?

1. Check the code that enforces these rules.
2. Check the code that allocates disk space, RAM; and user-authentication code.
3. Check all other kernel code.

Bugs anywhere in kernel can override these rules.

Memory protection doesn’t apply; language (C) doesn’t compensate.

The code we have to check is the trusted computing base.

Security metric #1: TCB size.

Eve can’t write Dan’s files unless there’s a TCB bug.

Eve’s actions: irrelevant.

Other software: irrelevant.

Millions of lines of code that we don’t have to check.

Do we need an audit log? No.

Keep computers separate? No.

Limit software Eve can run? No.

File sharing

So far have described complete user isolation.

But users want to share many of their files: consider the Web, email, etc.

I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

File sharing

So far have described complete user isolation.

But users want to share many of their files: consider the Web, email, etc.

I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.
Assume the hardware works. How do we verify that Eve can’t write Dan’s files?  
1. Check the code that enforces these rules. 
2. Check the code that allocates disk space, RAM; and user-authentication code. 
3. Check all other kernel code. Bugs anywhere in kernel can override these rules. 
Memory protection doesn’t apply; language (C) doesn’t compensate. 

The code we have to check is the trusted computing base. Security metric #1: TCB size. 

Eve can’t write Dan’s files unless there’s a TCB bug. 

Eve’s actions: irrelevant. Other software: irrelevant. 

Millions of lines of code that we don’t have to check. 

Do we need an audit log? No. 
Keep computers separate? No. 
Limit software Eve can run? No. 

File sharing 
So far have described complete user isolation. 
But users want to share many of their files: consider the Web, etc. 

I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.
The code we have to check is the **trusted computing base**. Security metric #1: TCB size.

Eve can’t write Dan’s files unless there’s a TCB bug.

Eve’s actions: irrelevant. Other software: irrelevant. Millions of lines of code that we *don’t* have to check.


File sharing

So far have described complete user isolation. But users want to share many of their files: consider the Web, email, etc. I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.
The code we have to check is the **trusted computing base**.
Security metric #1: TCB size.
Eve can’t write Dan’s files unless there’s a TCB bug.
Eve’s actions: irrelevant.
Other software: irrelevant.
Millions of lines of code that we *don’t* have to check.

Do we need an audit log? No.
Keep computers separate? No.
Limit software Eve can run? No.

---

**File sharing**
So far have described complete user isolation.
But users want to share many of their files: consider the Web, email, etc.
I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.
The code we have to check is the trusted computing base. Security metric #1: TCB size. Eve can’t write Dan’s files unless there’s a TCB bug. Other software: irrelevant. Millions of lines of code we don’t have to check. Do we need an audit log? No. Keep computers separate? No. Limit software Eve can run? No.

File sharing

So far have described complete user isolation. But users want to share many of their files: consider the Web, email, etc. I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

Say Frank creates a file, makes it readable to me. I save a copy. Later I look at the copy. Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source. Computer has to tell me that Frank was the source. I own the copy but Frank is the source.
The code we have to check is the trusted computing base.

Security metric #1: TCB size.

Eve can't write Dan's files unless there's a TCB bug.

Other software: irrelevant.

Millions of lines of code we don't have to check.

Do we need an audit log? No.

Keep computers separate? No.

Limit software Eve can run? No.

File sharing

So far have described complete user isolation.

But users want to share many of their files: consider the Web, email, etc.

I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

Say Frank creates a file, makes it readable to me.

I save a copy.

Later I look at the copy.

Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.

⇒ Computer has to tell me that Frank was the source.

I own the copy but Frank is the source.
The code we have to check is the trusted computing base.

Security metric #1: TCB size.

Eve can’t write Dan’s files unless there’s a TCB bug.

Eve’s actions: irrelevant.

Other software: irrelevant.

Millions of lines of code that we don’t have to check.

Do we need an audit log? No.

Keep computers separate? No.

Limit software Eve can run? No.

---

File sharing

So far have described complete user isolation.

But users want to share many of their files: consider the Web, email, etc.

I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

Say Frank creates a file, makes it readable to me.

I save a copy.

Later I look at the copy.

Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.

⇒ Computer has to tell me that Frank was the source.

I own the copy but Frank is the source.
File sharing
So far have described complete user isolation.

But users want to share many of their files: consider the Web, email, etc.

I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

Say Frank creates a file, makes it readable to me.

I save a copy.

Later I look at the copy.

Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.

⇒ Computer has to tell me that Frank was the source.

I own the copy but Frank is the source.
So far have described complete user isolation. But users want to share many of their files: consider the Web, email, etc. I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

Say Frank creates a file, makes it readable to me. I save a copy. Later I look at the copy. Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.

⇒ Computer has to tell me that Frank was the source. I own the copy but Frank is the source.

Obvious implementation: The OS kernel tracks source for each file, process. When my copying process opens the file from Frank, the OS kernel marks Frank as a source. When process creates file, the kernel copies source. Typical OS kernels today don’t even try to do this.
So far have described complete user isolation. But users want to share many of their files: consider the Web, email, etc.

I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

Say Frank creates a file, makes it readable to me. I save a copy.

Later I look at the copy. Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.

⇒ Computer has to tell me that Frank was the source.

I own the copy but Frank is the source.

Obvious implementation:
The OS kernel tracks source for each file, process. When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process. When process creates the file, the kernel copies source.

Typical OS kernels today don’t even try to do this.
So far have described complete user isolation. But users want to share many of their files: consider the Web, email, etc. I want to be able to mark a file I own as readable to just me; or also readable to Frank; or to Eve+Frank; or to a bigger group; or to the general public.

Say Frank creates a file, makes it readable to me. I save a copy. Later I look at the copy. Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file. \( \Rightarrow \) Computer has to tell me that Frank was the source. I own the copy but Frank is the source.

Obvious implementation: The OS kernel tracks source for each file, process. When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process. When process creates file, the kernel copies source. Typical OS kernels today don’t even try to do this.
Say Frank creates a file, makes it readable to me.

I save a copy.

Later I look at the copy.

Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file. 

⇒ Computer has to tell me that Frank was the source.

I own the copy but Frank is the source.

Obvious implementation:

The OS kernel tracks source for each file, process.

When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process.

When process creates file, the kernel copies source.

Typical OS kernels today don’t even try to do this.
Say Frank creates a file, makes it readable to me.
I save a copy.
Later I look at the copy.
Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.
Computer has to tell me that Frank was the source.
I own the copy but Frank is the source.

Obvious implementation:
The OS kernel tracks source for each file, process.
When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process.
When process creates file, the kernel copies source.
Typical OS kernels today don't even try to do this.

More complicated example:
Eve and Frank create files, make them readable to me.
I have a process that reads the file from Eve, reads the file from Frank, creates an output file.
Say Frank creates a file, makes it readable to me.
I save a copy.
Later I look at the copy.
Remember integrity policy #1: Whenever the computer shows me a file, it also tells me the source of the file.
Computer has to tell me that Frank was the source.
I own the copy but Frank is the source.

Obvious implementation:
The OS kernel tracks source for each file, process.
When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process.
When process creates file, the kernel copies source.
Typical OS kernels today don’t even try to do this.

More complicated example:
Eve and Frank create files, make them readable to me.
I have a process that reads the file from Eve, reads the file from Frank, creates an output file.

Obvious implementation:
The OS kernel tracks source for each file, process.

When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process.

When process creates file, the kernel copies source.

Typical OS kernels today don’t even try to do this.

More complicated example: Eve and Frank create files, make them readable to me.

I have a process that reads the file from Eve, reads the file from Frank, creates an output file.
Obvious implementation:
The OS kernel tracks
source for each file, process.
When my copying process
opens the file from Frank,
the OS kernel marks Frank
as a source for that process.
When process creates file,
the kernel copies source.
Typical OS kernels today
don’t even try to do this.

More complicated example:
Eve and Frank create files,
make them readable to me.
I have a process that
reads the file from Eve,
reads the file from Frank,
creates an output file.
Obvious implementation:
The OS kernel tracks
source for each file, process.
When my copying process
opens the file from Frank,
the OS kernel marks Frank
as a source for that process.
When process creates file,
the kernel copies source.
Typical OS kernels today
don't even try to do this.

More complicated example:
Eve and Frank create files,
make them readable to me.
I have a process that
reads the file from Eve,
reads the file from Frank,
creates an output file.

Integrity policy #1 ⇒
The OS kernel marks
both Frank and Eve
as sources for the process,
then sources for the file.
Obvious implementation:
The OS kernel tracks source for each file, process. When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process. When process creates file, the kernel copies source. Typical OS kernels today don't even try to do this.

More complicated example: Eve and Frank create files, make them readable to me. I have a process that reads the file from Eve, reads the file from Frank, creates an output file.

Integrity policy #1 ⇒ The OS kernel marks both Frank and Eve as sources for the process, then sources for the file.

Web browsing
Frank posts news-20140710 on his web server.
My browser retrieves the file, shows it to me.

Integrity policy #1 ⇒ My computer tells me that Frank was the source.

A modern browser tries to enforce this policy. But browser is a massive TCB, very expensive to check, full of critical bugs.
Obvious implementation:
The OS kernel tracks source for each file, process. When my copying process opens the file from Frank, the OS kernel marks Frank as a source for that process. When process creates file, the kernel copies source. Typical OS kernels today don't even try to do this.

More complicated example: Eve and Frank create files, make them readable to me. I have a process that reads the file from Eve, reads the file from Frank, creates an output file.

Integrity policy #1 ⇒ The OS kernel marks both Frank and Eve as sources for the process, then sources for the file.

Web browsing
Frank posts news-20140710 on his web server. My browser retrieves the file, shows it to me.

Integrity policy #1 ⇒ My computer tells me that Frank was the source.

A modern browser tries to enforce this policy.

But browser is a massive TCB, very expensive to check, full of critical bugs.
Obvious implementation:
The OS kernel tracks
source for each file, process.
When my copying process
opens the file from Frank,
the OS kernel marks Frank
as a source for that process.
When process creates file,
the kernel copies source.
Typical OS kernels today
don't even try to do this.

More complicated example:
Eve and Frank create files,
make them readable to me.
I have a process that
reads the file from Eve,
reads the file from Frank,
creates an output file.
Integrity policy #1 ⇒
The OS kernel marks
both Frank and Eve
as sources for the process,
then sources for the file.

Web browsing
Frank posts news-20140710
on his web server.
My browser retrieves the file,
shows it to me.
Integrity policy #1 ⇒
My computer tells me that
Frank was the source.
A modern browser tries
to enforce this policy.
But browser is a massive TCB,
very expensive to check,
full of critical bugs.
More complicated example:
Eve and Frank create files, make them readable to me.
I have a process that reads the file from Eve, reads the file from Frank, creates an output file.
Integrity policy #1
The OS kernel marks both Frank and Eve as sources for the process, then sources for the file.

Web browsing
Frank posts news-20140710 on his web server.
My browser retrieves the file, shows it to me.
Integrity policy #1 ⇒ My computer tells me that Frank was the source.
A modern browser tries to enforce this policy. But browser is a massive TCB, very expensive to check, full of critical bugs.
More complicated example:
Eve and Frank create files, make them readable to me.
I have a process that reads the file from Eve, reads the file from Frank, creates an output file.

Integrity policy #1
The OS kernel marks both Frank and Eve as sources for the process, then sources for the file.

Web browsing
Frank posts news-20140710 on his web server.
My browser retrieves the file, shows it to me.

Integrity policy #1 ⇒
My computer tells me that Frank was the source.

A modern browser tries to enforce this policy.
But browser is a massive TCB, very expensive to check, full of critical bugs.

What if I instead give Frank a file-upload account on my computer?
Frank logs in, stores a file news-20140710.
I start a process that looks at the file.

If OS tracks sources then it tells me that Frank was the source.
Web browsing

Frank posts news-20140710 on his web server.
My browser retrieves the file, shows it to me.

Integrity policy #1 ⇒
My computer tells me that Frank was the source.

A modern browser tries to enforce this policy.
But browser is a massive TCB, very expensive to check, full of critical bugs.

What if I instead give Frank a file-upload account on my computer?
Frank logs in, stores a file news-20140710. I start a process that looks at the file.

If OS tracks sources then it tells me that Frank was the source.
Web browsing

Frank posts `news-20140710` on his web server.
My browser retrieves the file, shows it to me.

Integrity policy #1 ⇒
My computer tells me that Frank was the source.

A modern browser tries to enforce this policy.
But browser is a massive TCB, very expensive to check, full of critical bugs.

What if I instead give Frank a file-upload account on my computer?

Frank logs in, stores a file `news-20140710`.
I start a process that looks at the file.
If OS tracks sources then it tells me that Frank was the source.
Web browsing
Frank posts news-20140710 on his web server.
My browser retrieves the file, shows it to me.

Integrity policy #1 ⇒
My computer tells me that Frank was the source.

A modern browser tries to enforce this policy.
But browser is a massive TCB, very expensive to check, full of critical bugs.

What if I instead give Frank a file-upload account on my computer?
Frank logs in, stores a file news-20140710.
I start a process that looks at the file.
If OS tracks sources then it tells me that Frank was the source.
Web browsing
Frank posts news-20140710 on his web server.
My browser retrieves the file, shows it to me.

Integrity policy #1 ⇒
My computer tells me that Frank was the source.

A modern browser tries to enforce this policy.
But browser is a massive TCB, very expensive to check, full of critical bugs.

What if I instead give Frank a file-upload account on my computer?
Frank logs in, stores a file news-20140710.
I start a process that looks at the file.
If OS tracks sources then it tells me that Frank was the source.

Why should this be manual?
Browser creates process that downloads news-20140710 from Frank's server.
("Creating a process is slow."
—Oh, shut up already.)
OS automatically adds URL as a source for the process.
Process shows me the file.
OS tells me the URL.
Frank posts news-20140710 on his web server. My browser retrieves the file, shows it to me.

Integrity policy #1

My computer tells me that Frank was the source.

A modern browser tries to enforce this policy.

But browser is a massive TCB, very expensive to check, full of critical bugs.

What if I instead give Frank a file-upload account on my computer?

Frank logs in, stores a file news-20140710.

I start a process that looks at the file.

If OS tracks sources then it tells me that Frank was the source.

Why should this be manual?

Browser creates process that downloads news-20140710 from Frank’s web server. (“Creating a process is slow.”—Oh, shut up already.)

OS automatically adds URL as a source for the process.

Process shows me the file.

OS tells me the URL.
What if I instead give Frank a file-upload account on my computer?

Frank logs in, stores a file news-20140710. I start a process that looks at the file.

If OS tracks sources then it tells me that Frank was the source.

Why should this be manual?

Browser creates process that downloads news-20140710 from Frank’s web server.

(“Creating a process is slow.” —Oh, shut up already.)

OS automatically adds URL as a source for the process.

Process shows me the file. OS tells me the URL.
What if I instead give Frank a file-upload account on my computer?

Frank logs in, stores a file `news-20140710`. I start a process that looks at the file.

If OS tracks sources then it tells me that Frank was the source.

Why should this be manual?

Browser creates process that downloads `news-20140710` from Frank’s web server.

("Creating a process is slow.
—Oh, shut up already.

OS automatically adds URL as a source for the process.

Process shows me the file. OS tells me the URL.

If I instead give Frank a file-upload account on my computer? Frank logs in, stores a file `news-20140710`. I start a process that looks at the file. If OS tracks sources then it tells me that Frank was the source.

Why should this be manual? Browser creates process that downloads `news-20140710` from Frank’s web server. (“Creating a process is slow.” —Oh, shut up already.) OS automatically adds URL as a source for the process. Process shows me the file. OS tells me the URL.

Closing thoughts Is the community even trying to build a software system with a small TCB that enforces integrity policy #1? If software security is a failure, does this mean that security is impossible, or does it mean that the community isn’t trying?
Why should this be manual?

Browser creates process that downloads news-20140710 from Frank’s web server.

(“Creating a process is slow.”—Oh, shut up already.)

OS automatically adds URL as a source for the process.

Process shows me the file.

OS tells me the URL.

Closing thoughts

Is the community even trying to build a software system with a small TCB that enforces integrity policy #1?

If software security is a failure, does this mean that security is impossible, or does it mean that the community isn’t trying?
Why should this be manual?

Browser creates process that downloads news-20140710 from Frank’s web server.

(“Creating a process is slow.” —Oh, shut up already.)

OS automatically adds URL as a source for the process.

Process shows me the file.
OS tells me the URL.

Closing thoughts

Is the community even trying to build a software system with a small TCB that enforces integrity policy #1?

If software security is a failure, does this mean that security is impossible, or does it mean that the community isn’t trying?
Why should this be manual?

Browser creates process that downloads news-20140710 from Frank’s web server.

(“Creating a process is slow.” —Oh, shut up already.)

OS automatically adds URL as a source for the process.

Process shows me the file.

OS tells me the URL.

Closing thoughts

Is the community even *trying* to build a software system with a small TCB that enforces integrity policy #1?

If software security is a failure, does this mean that security is impossible, or does it mean that the community isn’t trying?