Thank You!

Security can be:
  Boring
  Frustrating
  Intimidating

But your interest today says you care about protecting your research.
Introductions

About This Presentation

Approachable
   Eight (short) breaks for questions

Actionable

Reasonable
Introductions (cont.)

About Myself

Senior Security Analyst at SDSC
Security for HPC environments since mid 2000s
Interested in the intersection of people and security
A Philosophical Journey Into Security

Security is a journey, not a destination.
"Why Me?"

Two distinct contexts:
First-time victims
Future victims

Often followed by "But...":
I have nothing interesting
I have no sensitive data
My research is public
"Why Me?" (Answered)

To see what you (don't) have, the attacker must first get in.

The attacker knows you have exactly what they want. (and it's probably not what you think it is)

There is no attacker. You just got a bad dice roll.
What Could Go Wrong?

Depends on circumstances!

Examples:
Data corruption / deletion.
Someone else publishes your work.
Integrity of your research is called in to question.
Additional stolen access used against partners and affiliates.
We all look bad and suffer loss of reputation.

Decisions / Policy made using research based on corrupted data.

Expensive!
Knowledge Is Power

... and unfortunately, attackers have the knowledge.

We can't predict each attacker's goals and motives.
We don't know what tools they have.
We don't know the extent of their resources.

But we still must keep them at bay.
A Tangent!

A head-on defense is very hard when you don't know what your adversary is up to.

Conventional wisdom: when you park at the mall, don't leave your bags in plain sight.

But what if the thief doesn't care about your shopping?
A Zen Moment
The last few slides were anxiety-inducing, however...

A successful attack (or bad dice roll) doesn't mean game over.

What matters are the consequences.
The power of "What if...?" and the mistakes of others allow us to anticipate many of them.

We can prepare now, knowing what may come later.
Questions?

Break 1 of 8

Up next: Security Responsibilities
Security is Everybody's Responsibility

Who's responsible for doing what?

Who's responsible for *not* doing what?
We’re In This Together

HPC Environment

Provided By SDSC
We're In This TOGETHER (cont.)
Managing Expectations

SDSC will enforce the permissions you set to protect your data from other users.

SDSC will protect your account from other users.

SDSC will provide access to the computational resources that you are entitled to.
Managing Expectations (cont.)

SDSC expects you to protect your account credentials to the extent you have control. (More later!)

SDSC expects you to protect your data to the extent you have control. (More later!)

SDSC expects you to only use its resources for the purpose you have been authorized to use them for.
Questions?

Break 2 of 8

Up next: Client Security
Pop Quiz!

What's the *most important* thing you can do to protect your account credentials?

Choose a password with numbers, letters, and symbols?
(No.)

Install and run anti-malware software?
(Second-most important!)

Keep your machine and software updated!
FOMO Isn't Just for Social Circles

Keep attackers from easily taking control of your machine by applying security updates to its operating system and other software.

Detect and stop many tools used by attackers by running *current* anti-malware software.

Your institution may already do this for you...

But what about your personal machines?
Devil or Angel?

SDSC cannot protect your computers!

An attacker on your computer can do anything you can (and sometimes more).

SDSC does its best to protect users from each other.

But only you can protect yourself.
How it Works: Opportunistic Attacks

(1) Use the computer normally.
How it Works: Opportunistic Attacks

(2) Un-patched programming flaws allow malware to run.
How it Works: Opportunistic Attacks

(3) Somebody / Something else has control*.

* Don't feel bad. This can happen even if you did everything right!
Dealing With Opportunistic Attacks

Updates!

Upgrade?

Update!! (and run anti-malware)
Questions?

Break 3 of 8

Up next: Logging In and Passwords
Crossing the Threshold: Logging In

Your first time logging in to SDSC's HPC environment will likely involve a password.

So... What makes a good password?

- Contains upper-case letters.
- Contains lower-case letters.
- Contains numbers.
- Contains symbols.
- Changed regularly.

Ref: NIST SP 800-63, Appendix A
Unusable Passwords

Old rules are *hostile to users*.

Hard to type.

Hard to remember!

Remembered? Great, time to change.

New guidance...
Usable Passwords

Current guidelines address usability.

No composition rules.

8-character minimum.

No arbitrary changes.

Check against lists of known-bad passwords.

Ref: NIST SP 800-63B, Section 5.1.1
Usable Passwords (cont.)

What this means for you:

Ref: https://xkcd.com/936/
Usable Passwords (cont. 2)

Additional guidance:

*Never* use the same password for different services.

Don't share your password with other people. This includes any kind of support staff, real or otherwise!

Consider using a password-manager program.

It may be okay to write your passwords down if they are kept physically secure. Your institution can provide guidance on complying with their policies.
Usable Passwords (cont. 3)

Don't share your password with other people, including support staff.

Don't store your password in any electronic form except for a password manager that encrypts its password list.

Examples of where to avoid storing passwords:
- In a file (including scripts)
- In an email draft
- In an Excel spreadsheet
- Taped to your monitor
Questions?

Break 4 of 8

Up next: Using SSH Keys Safely
Welcome Back!

You've changed your password, but it's still time-consuming to use.

SSH Keys to the rescue!

Maybe?
SSH Keys

The key fingerprint is:

First it converts the key to a passphrase:
schacon@mylaptop

It's up to you whether to type a password or add the -o option: it's up to you.

For the key when prompted on it: Enter the password you wish to use or continue without a password. Press enter twice to continue without asking for a password.

Private key should be able to be locked automatically. It should be able to be unlocked automatically. It should be able to be unlocked automatically.

SDSC SUPERCOMPUTER CENTER

UC San Diego
SSH Keys: Background

"A Message"  

bf25457a69ebefde74972265153ee664d1ceca814ce1e77f8

"A Message"
"Isn't that like a password?"

"Didn't you say not to store passwords electronically unless encrypted?"
IF YOU COULD PUT A PASSPHRASE ON YOUR PRIVATE KEYS.

THAT WOULD BE GREAT.
SSSHH KKeys: Safety First!

Keep your friends close, and your private keys closer! (Only on the computer you're physically touching.)

Not on SDSC's HPC systems.

Use a long passphrase. 24+ characters long.

It's okay to use the same key pair everywhere... Only as long as you protect the private key.
SSH Keys: Unusable?

"I have to type THAT passphrase every time to log in!"
"How's that better than a password?"

"What if I need to log in to something else from the HPC system?"

"This must be really secure. It's a pain to use."
SSH Keys: Don't Panic!

SSH Agent to the rescue!

Securely stores your private key in decrypted form.

SSH client communicates with agent.

Communication with agent can be forwarded or proxied. Appears as if private key were available on the server.
SSH Keys: What It Looks Like
SSH Keys: Step 1 – Generate Key Pair

Only need to do once.
(Unless you lose [control of] your private key)

Skip this if you already have a protected private key.
Generate a new one if your private key has no passphrase or just a password.

Key pairs have different formats. Use this one for compatibility.
Type: RSA
Bits: 2048 or more
SSH Keys: Generate Key Pair (Mac/Linux)

Software is built-in

Protect your key with a long passphrase!
SSH Keys: Generate Key Pair (Windows)

Puttygen

You will need the public key. Remember to save a copy.

Protect your key with a long passphrase.

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
SSH Keys: Step 2 – Add to Agent

Only need to do this when agent does not have your key.

Agent will expunge your key upon reboot.*
Or when you remove it.

To minimize opportunity for misuse, remove your key from the agent when you’re done for the day.
Mac and Linux support automatic removal.
Windows does not.

* Windows has a native ssh-agent. It will keep your key even after rebooting. I do not recommend it!
SSH Keys: Add to Agent (Mac/Linux)

Software is built-in
SSH Keys: Add to Agent (Windows)

pageant

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
SSH Keys: Step 3 – Deploy Public Key

The servers you connect to will need to know your public key.

Your public key goes in to ~/.ssh/authorized_keys

Only add your public keys!
Fewer is better.

Each key is one (very long) line.
SSH Keys: Deploy Public Key (Mac/Linux)

Software is built-in

```
ssakai@mnemeth:~$ ssh-copy-id scott.sdsc.edu
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed - if you are prompted now it is to install the new keys
ssakai@scott.sdsc.edu's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'scott.sdsc.edu'"
and check to make sure that only the key(s) you wanted were added.
```

ssakai@mnemeth:~$
SSH Keys: Deploy Public Key (Windows)

Manual process.

Copy public key from puttygen.

Log in to host.

Edit ~/.ssh/authorized_keys
Paste!
Save!
Remember: One key per line
SSH Keys: Step 4 – Just Use It

Your ssh client will automatically use the agent.

“Agent Forwarding” may be disabled by default. (It should be.)

An attacker on the server you log in to can also use your agent when you connect with agent forwarding turned on.

It’s okay to use agent forwarding when connecting to hosts you trust. (But should only be used when you expect to “hop through”)

SSH Keys: Just Use It (Mac/Linux)

Software is built-in
SSH Keys: Just Use It (Windows)

Putty

Controls for agent forwarding are a little buried.

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
Questions?

Break 5 of 8

Up next: Protecting Your Account (continued)
I’m In, Now What?

Your ability to protect your account doesn’t stop here!

About running network services.

Copy-paste cautions.

Sudo.
A network service is a program that performs actions as a user in response to requests sent over the network.

SDSC discourages user-run network services on HPC.

Please reach out to SDSC’s Support staff if you intend to run a service such as these:
Cutting Corners With Cut and Paste

Cut-and-paste instructions are common. Caution!

The commands to cut/paste may not do what the instructions say they do.

The site can instruct your browser to replace the contents of the clipboard with something dangerous.

To be sure: paste in a text editor first.
Ask Consultants for Help; Su-don’t

SDSC does not grant privileged access to users of its HPC systems.

Instructions involving `sudo` will not work!

Please contact SDSC’s consultants if you are following instructions involving the `sudo` command.

The software you need may already be there.
Questions?

Break 6 of 8

Up next: Protecting Your Data
What About My Data?

You control your data!

Sensitive data isn’t allowed.

File permissions for everyone.

Finding corruption.
A Touchy Subject

What makes data sensitive?
Regulatory compliance
Privacy or financial implications
Contractual obligations

SDSC’s HPC systems are not designed to protect sensitive data.

Everybody will lose if there’s a breach involving your sensitive data.

Please do not bring sensitive data into these environments.
An Access Control Freak-Out

$ chmod 777 .

Fixed?

NO.
We All Make Mistakes…

Someone else’s mistake can become your mistake.

`chmod 777` sets “world writable” permissions
ANYONE can make changes!

See the oops?

Your files would be deleted!
(Not everyone has seen this presentation. Please don’t try this command.)
Ack-ing Access Controls

Permissions are broken down into three major categories:
User
Group
Other
Ack-ing Access Controls (cont.)

Each category has three permissions
Read (4)
Write (2)
Execute (1)
Ack-ing Access Controls (cont. 2)

To pick numeric chmod permissions:
Each digit corresponds to permission category
[User] [Group] [Other]

The value of each digit is the sum of desired permissions
[Read (4)] [Write (2)] [Execute (1)]

“User can read, write; Group can read; Other can do nothing”
(4 + 2) (4) (0)
chmod 640
It’s a Group Thing

When you need to grant write access to a collaborator, use group permissions to grant write access, instead of “other”.

The `chown` command can change group ownership to any group you are a member of.

```
chown :groupname
```

Pick a group common to you and your collaborator. The `id -a` command lists the groups for each user.

```
id -a ssakai
```
Granting Group-write Permissions
A Minor Omission on Permissions

In the previous example, files and directories created in the mtcollab directory may be owned by a different group besides sds154.

There’s an additional digit to the left of the chmod user permissions, for attributes. If omitted no attributes are changed.

Set-UID (4)
Set-GID (2)
Sticky (1)

The chmod permissions in the previous example should be 2770 to avoid this problem.
What About Default Permissions?

In the example, files and directories created in the *mtcollab* directory might be missing the group-write bit, or worse, have *chmod permissions 777*.

Default permissions are set by *umask*.

The umask settings persist for the duration of your login and are inherited by any processes you run.
What About Default Permissions? (cont.)

The `umask` permissions are very similar to the `chmod` permissions. Start with “all permissions granted to all”. Any permissions present in the `umask` are removed.

Typical umask values:
“By default: user can do all; group, other can only read (and execute)”
`umask 0022`
“By default: user and group can do all; other can only read (and execute)”
`umask 0002`
“By default: user and group can do all; other has no permission”
`umask 0007`
Questions?

Break 7 of 8

Up next: Data Corruption
Detecting Data Corruption

While SDSC uses technologies that help prevent or detect data corruption, it is still possible to end up processing corrupted data.

Consider including integrity checks as part of your workflow.

For applications that do not inherently include integrity checking, try one of the digest tools mentioned next.
Getting In Digestion

Cryptographic digests take arbitrary data and produce a substantially smaller fixed-size sequence of bytes called a digest.

The digest is reasonably unique for every input.

Small changes cause big changes to the digest.

This makes them ideal for detecting corruption.

Some tools: sha256sum, sha1sum, md5sum (shasum and md5 on Mac)
Questions?

Break 8 of 8

Up next: Closing Remarks
Closing Remarks

This is not a comprehensive presentation! Awareness is the goal.

Security is a complex domain that cannot be condensed into a slide deck.

Develop a security-focused frame of mind.

It’s not paranoia, it’s preparedness.
Thank You!

Feedback to follow if time permits!