Hello everyone! Welcome to Zero to Secure where I'll tell you today about how we secured our Mac environment without "much" Pain

John Mahlman

Mac Engineering Lead, Leidos

@jmahlman

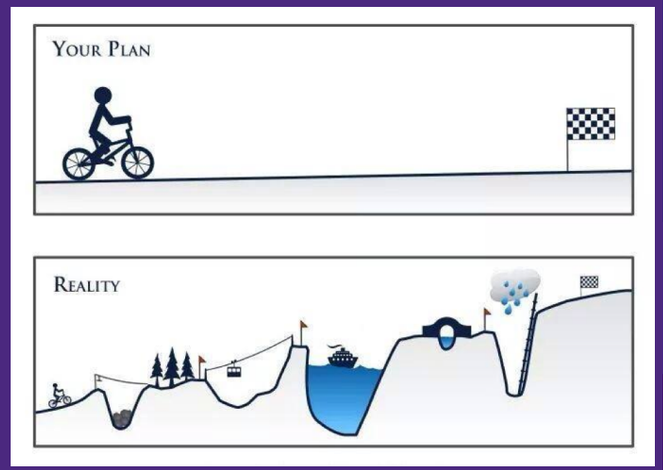
https://yearofthegeek.net

I am your speaker, John Mahlman. I've been a Mac admin for about 13 years now and I'm currently the Mac Engineering Lead for Leidos.

Leidos is an international technology and research contractor and one of the top 5 DoD contractors in the US.

Let’s set expectations

What this presentation is *not…*

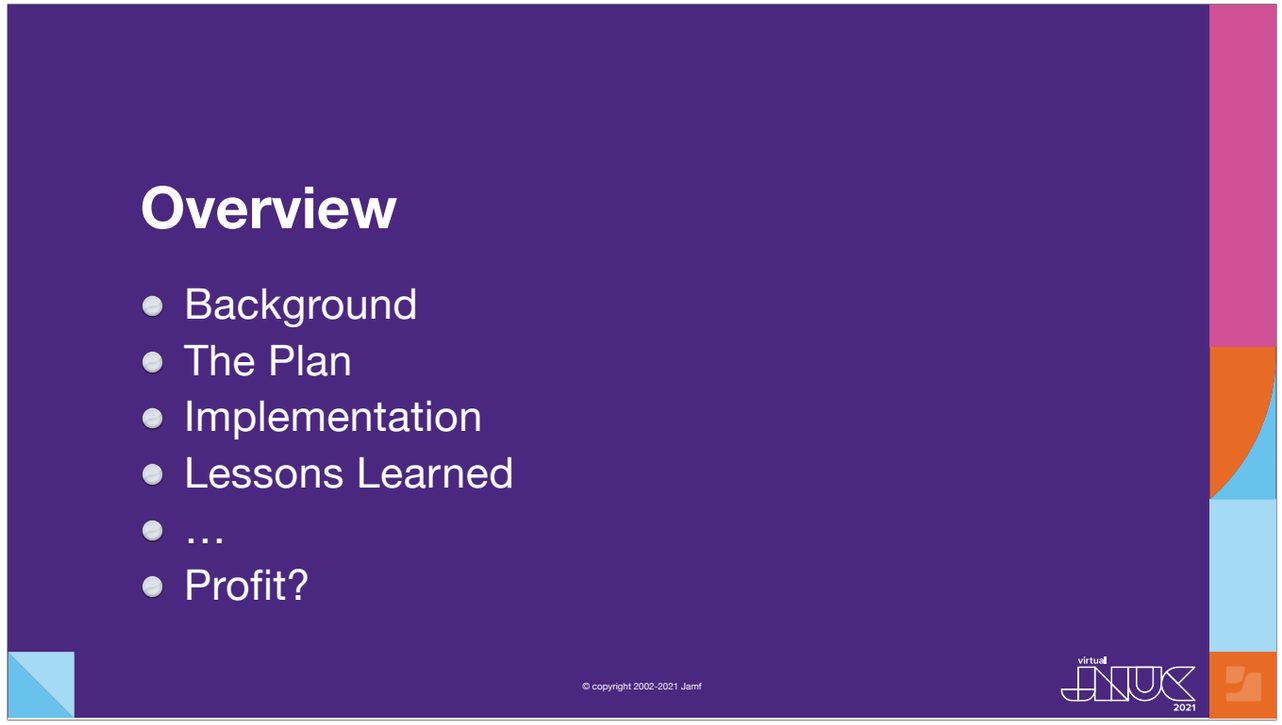


Before I start, I want to set some expectations about what this presentation is and what it is not.

This is not a highly technical presentation. I *will* share some technologies that we used to get our devices up to compliance and some suggestions on scripts and tools, but I will not be taking a huge deep dive into them.

This is more of a project management presentation; I will discuss how we planned the roll out and how we communicated it to users.

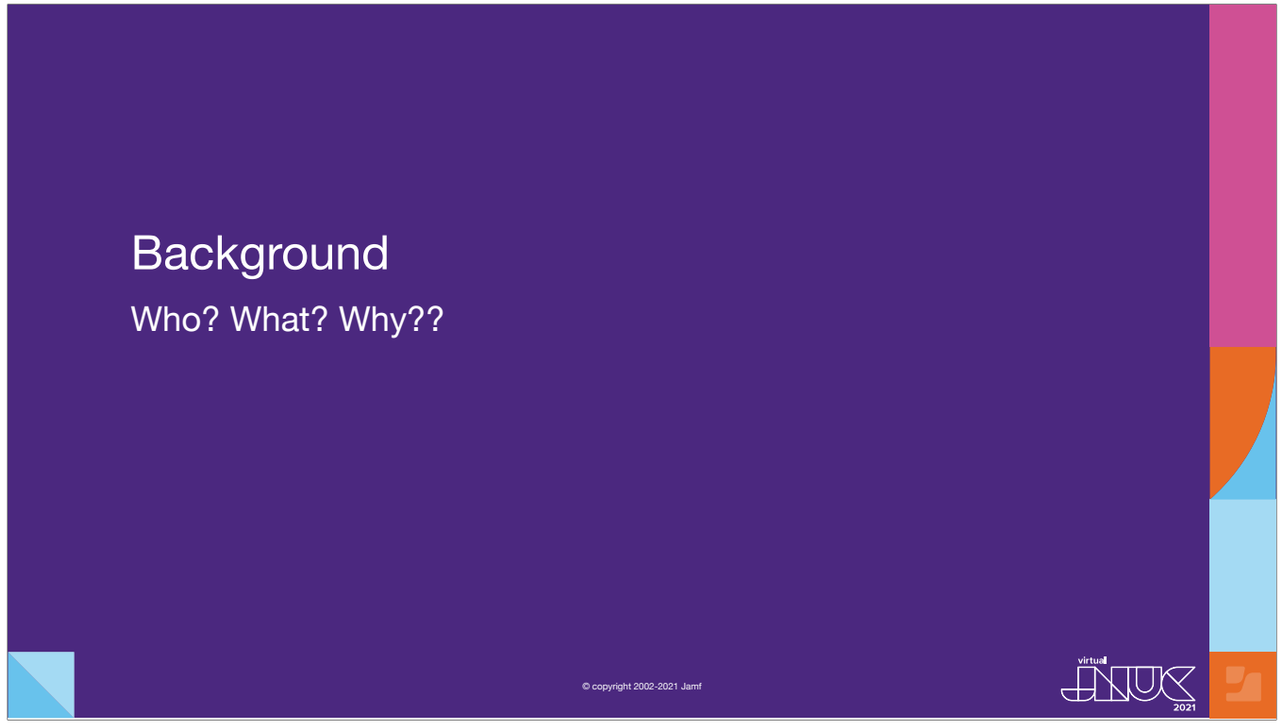
So, if you were hoping I’d give you a big script to get compliant, I’m sorry. You still have work to do :)



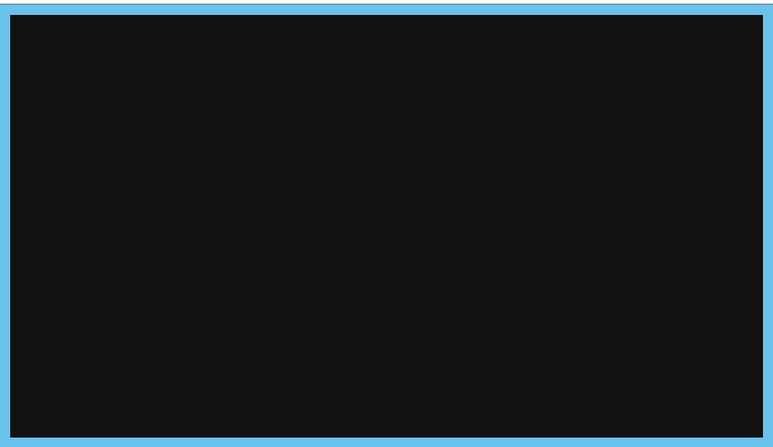
Now that that’s out of the way, here’s a very brief overview of the presentation.

I’ll go over some background; why we’re doing this, what we’re doing, and how we got here.

The planning stages..Implementation of the plan, some lessons learned…and whatever else!



Okay, let’s start with some background on the project in general so you can get a scope of work in your mind.



End Goal…

Ensure all Macs are managed and secure.

Simple goal, ensure all Macs are managed and secure. We needed to do this for various reasons which I'll get into
:click:
but basically, we just needed the thumbs up from cybersecurity for all Macs in the environment.

Background - Why

- ~400-700 total Macs in the environment
  - ~100 Enrolled in Jamf already
- Didn't know exactly where the devices were/who had them
- Need to secure devices for compliance **within ~~12~~ 8 months**
  - NIST 800-171
  - CMMC level 3/4
- Compliance required to win contracts
- Make life easier for users and admins

Speaking of all macs…we believed we had anywhere from 4-700 macs in the wild with around 100 in jamf already

Now, we didn't know who had the machines, but we did have some ways to find them (more on that later)
We were asked to secure with NIST 800-171 with the eventual goal of CMMC level 3 or 4 :**click**: within 12 months!
If you're not familiar with CMMC, it's The Cybersecurity Maturity Model Certification. It is a unified standard for implementing cybersecurity across DoD and other gov't information systems. It's pretty new, 1.0 was releases in January 2020
CMMC level 3 is basically NIST 800 with some added standards. Level 4 is where things get much more difficult, deep discussion is not in scope of this presentation but for more info I'll include some links at the end.
We needed this compliance to win contracts. We're a defense contractor, higher compliance means more contracts means more business
We also wanted to do this to make life a lot easier for user and admins
Oh wait, I said 12 months? I meant 8 months….

Background - Why

- Enterprise adding more controls for security
  - MFA
  - Application Controls
  - Certificate Verification
  - Blood samples
  - Retina scans
  - PIV Tokens

With endpoint compliance comes infrastructure compliance
The enterprise was already implementing these controls…I'll let you narrow down which ones are real!
Managing these on unmanaged devices is a nightmare for everyone, so getting control of all of the devices allows for us to standardize but also meet those security requirements
It's a win-win for us and users….
Mind you, these users have been around for a while, it's not like the Macs are a new phenomenon, they were just ignored…

Background - How

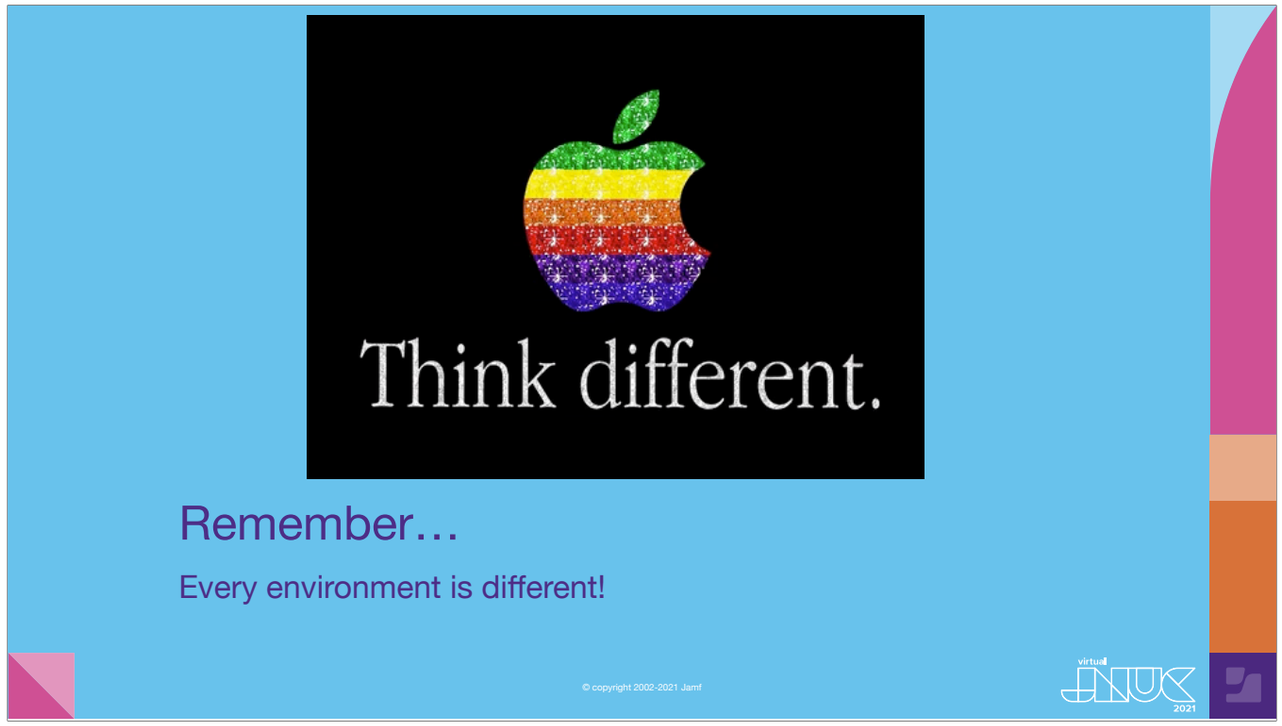
- Legacy practices
- Legacy tools
- Macs not “in scope” for anything
- No enterprise-level support
- Users “fending for themselves”
- Business groups making their own compliance guidelines
- Animosity toward IT

And some quick background on how we got here…and what it caused…
Some of these go hand in hand, right? Legacy practices and tools….and when they replace those old tools they get something not mac friendly
:click:
This led to things that you’d expect, users and groups fending for themselves to make compliant devices and the lack of support and the introduction of new tools that are not mac friendly led to them just not liking IT and not trusting us

The Process

Make a plan, communicate it, do it!

Now that we have some background, we had to produce a process to deploy things.

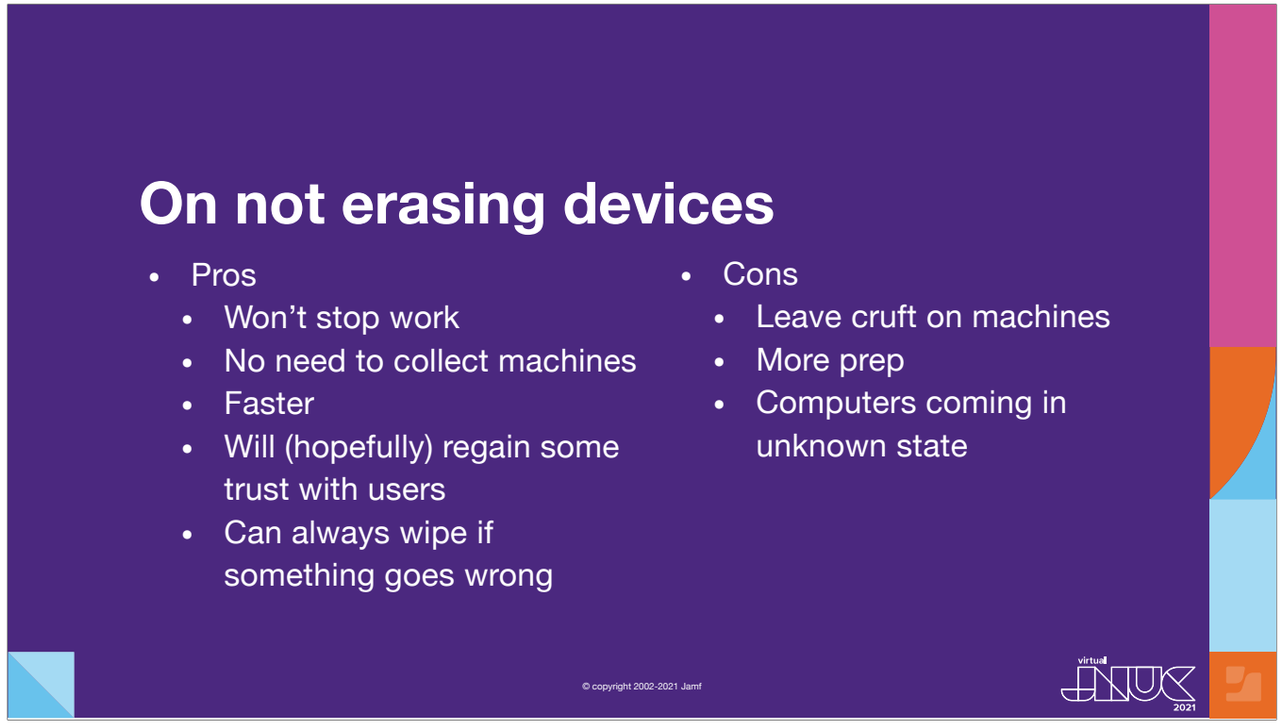


Remember, every environment is different. What I talk about here may not work for you but I hope this at least gives you direction.

Things to keep in mind…

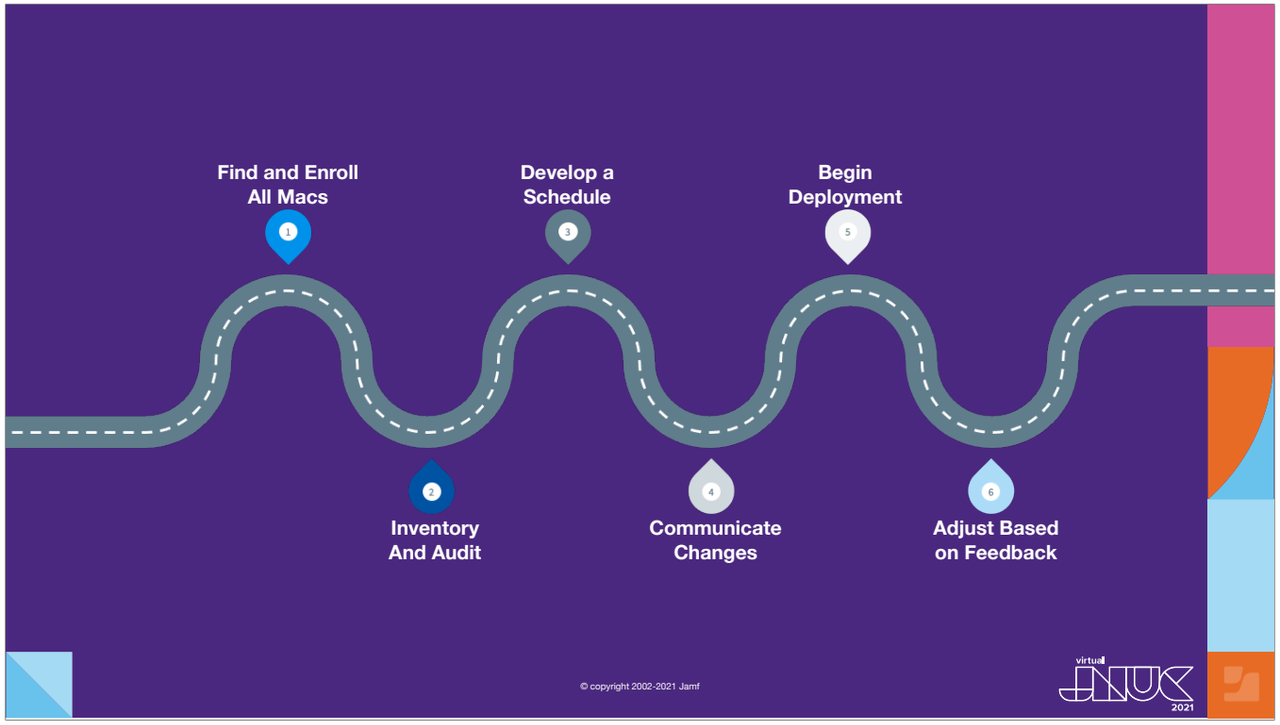
- Most Mac users are developers
  - Pretty tech savvy
  - Doing development for customers (making $$$ for business)
- Don’t break the business
- Secure first, enhance after
- Not erasing devices

Some things that we had to keep in mind based on the information we gathered
First, most of our Mac users are actually developers, they’re fairly tech savvy and they’re doing the work that keeps the company making money
Because of this, we didn’t want to break the business, if devs are not able to use their computers, they don’t deliver products to customers
Our idea was to secure the devices first and then enhance things later. There will be plenty of time to enhance the service after it’s secure…
Based on this info..we decided that :click: we’re not erasing the Macs



This was a bit controversial internally, but we took the info from the last slide and made some quick pros and cons

I’ll just give you a moment to read this, this is all things you probably know…we thought about it a good deal actually and my colleague and I both had different approaches…but I swayed him :)



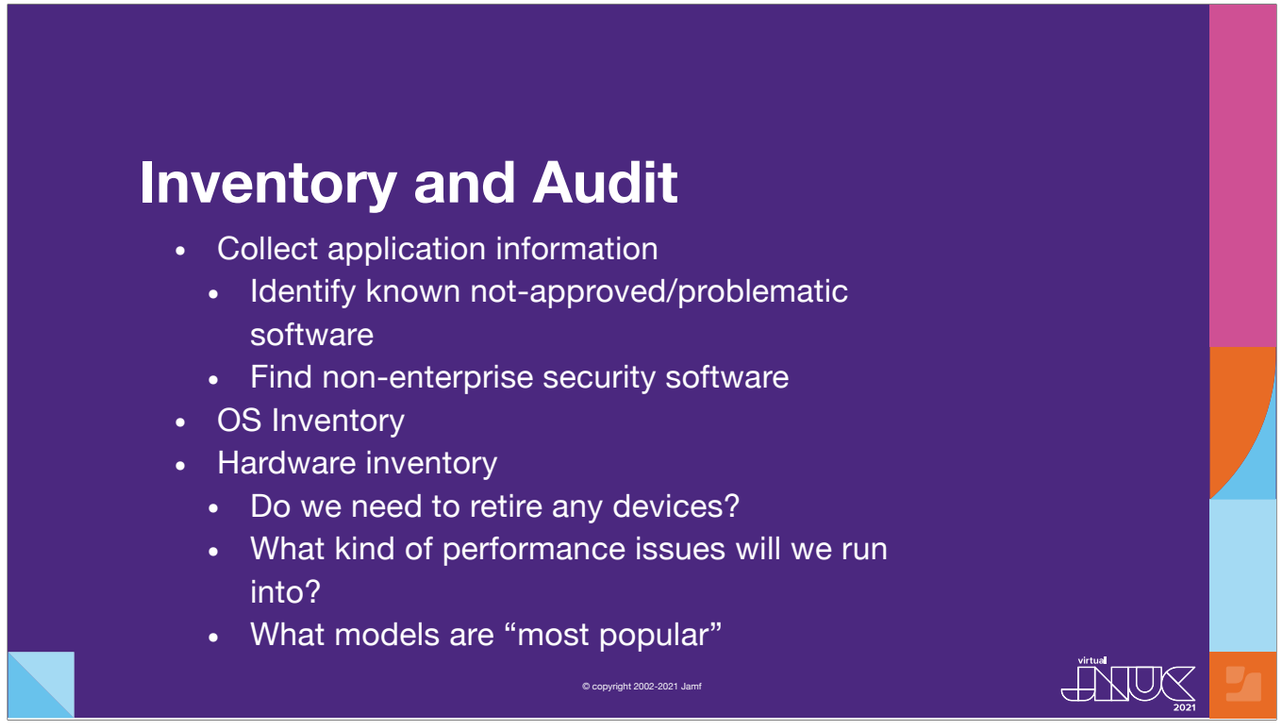
Now that we made some early decisions, we made a roadmap.

Collect and inventory machines, develop the schedule and tell the users and then deploy and adjust! Of course this is overly simplified..

Find and Enroll all Macs

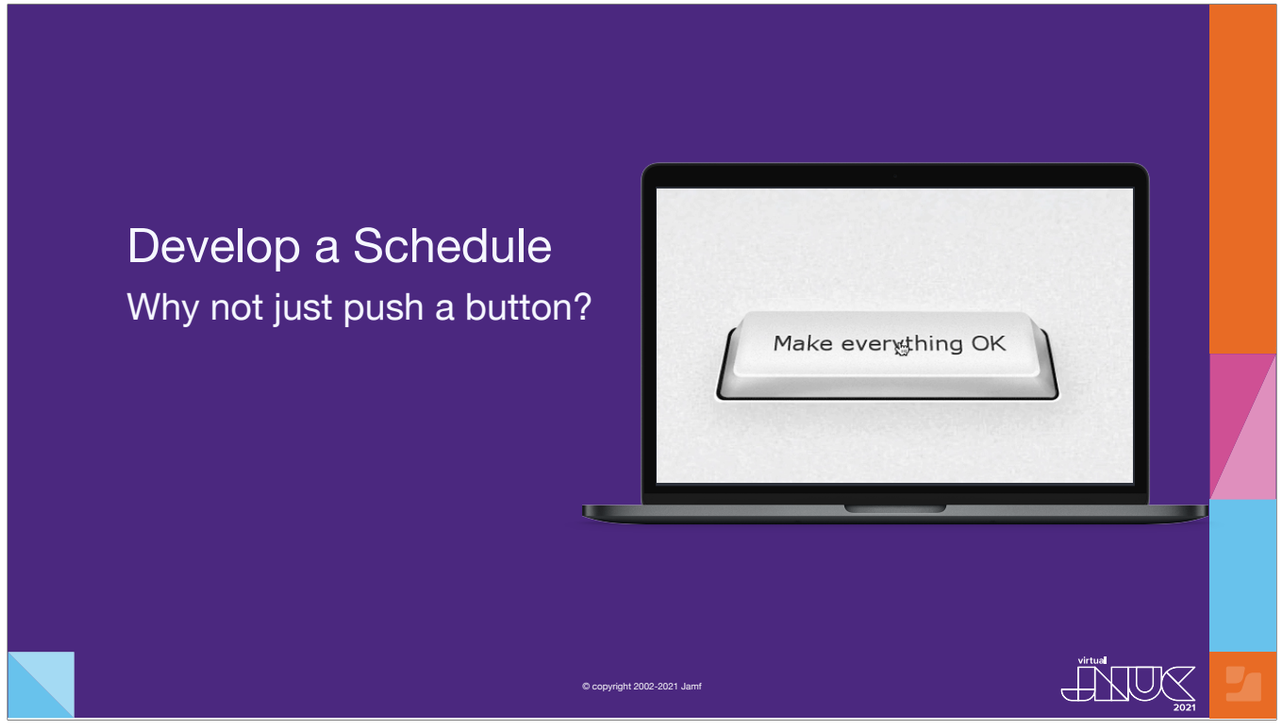
- Used various methods to find the device
  - Apple Business Manager
  - Purchase Records
  - VPN Logs
  - Anti-Virus Logs
  - Tickets
- Used various methods to enroll the devices
  - Blog/Forum/Newsletter Posts
  - Direct Emails to known users
  - Push with Anti-Virus server

The first step was just to find the machines. We used a few method to find them…
Thankfully we had good purchase records and already had ABM, so we knew about most machines…
But we still needed to find who had them so we can contact them, we used infrastructure logs
We had users already using VPN and AV in some cases but still weren't managed, so we got their info that way and used the various methods there to enroll them
Most of the enrollments were user-initiated but we had some enroll by trying to push an enrollment package via our AV server



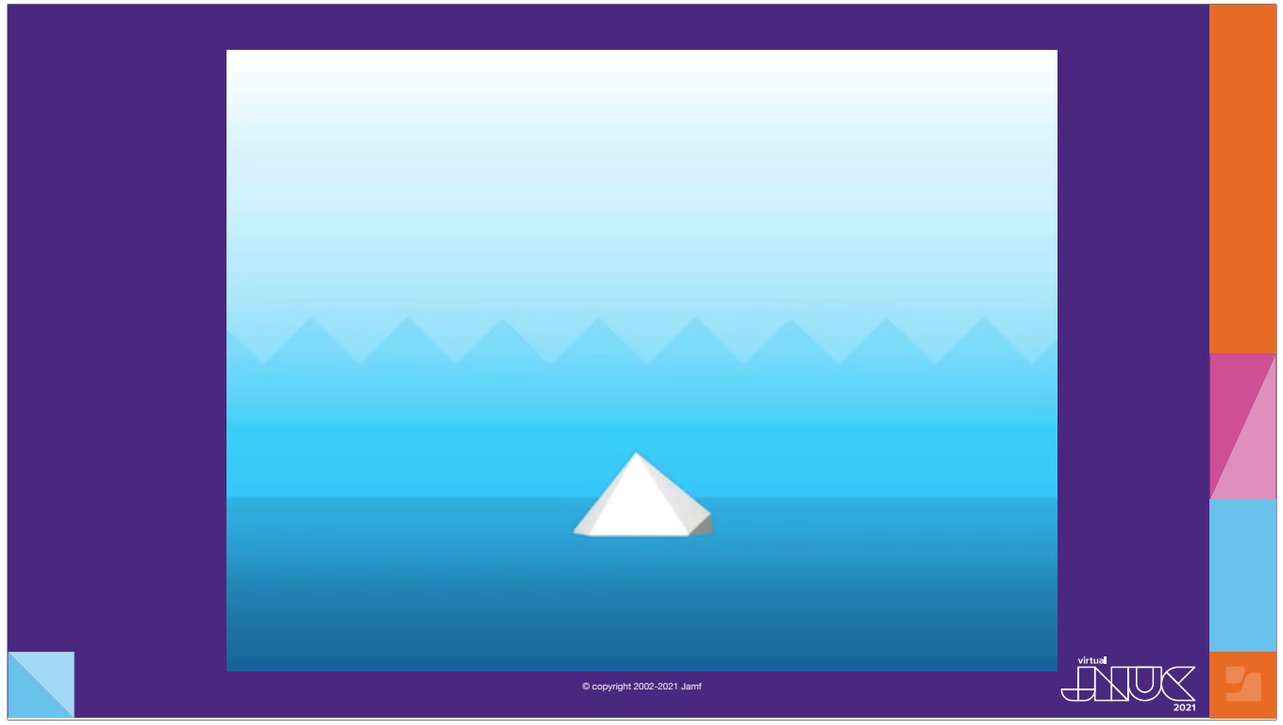
Once we started to get the machines in, we started inventory of software and hardware
We needed to ensure the software was okay to keep on the systems and sort of more importantly, we had to ensure we were not going to install another security suite on top of an existing suite
We wanted to see how the OS breakdown was, we only wanted to support Mojave and Catalina but we had systems with 10.13 and a few with 10.12, so we had to figure that out as well

As for hardware, we wanted to make sure the machines were ”good enough” to run our security stack but also ensure that when we manage them the performance hit wouldn’t be too bad
We also wanted to make sure the systems can run 10.14 at a minimum and if not, we’d retire them
Lastly, we wanted to see what was the “most popular” models
We would use this to determine which machines we would offer for purchase in our hardware catalog (spoiler, 15-inch MBP was the most popular by a long shot)



Alright, we have the machines enrolling, we have inventory being analyzed, now we wanted to figure out how to deploy the security settings and software updates!

We knew the baseline we were using, we had almost all of the software we were going to use chosen already, why not just push a button to make everyone compliant? Well, we could do this for new devices, but those existing ones…

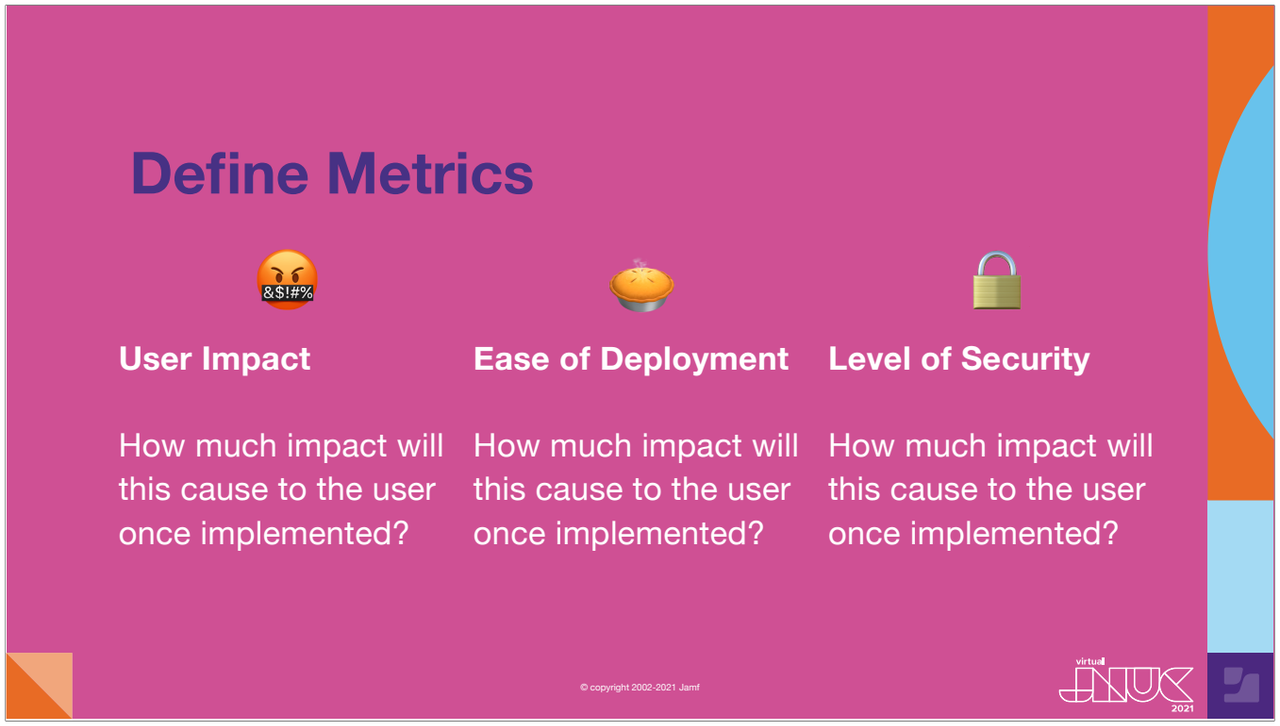


Think of an iceberg. We see the tip of the iceberg and we know there's a little bit underneath but we’re really not sure how much is hiding there.

We push that button :click: and suddenly the entire thing shows up at one time and now, we’re under water. We know ice bergs can sink ships and we didn’t want to have that happen.

	A	B	C	D	E
1	section #	recommendation #	title	status	assessment status
2	1		Install Updates, Patches and Additional Security Software	published	
3	1	1.7	Computer Name Considerations.	published	Manual
4	2		System Preferences	published	
5	2.1		Bluetooth	published	
6	2.2		Date & Time	published	
7	2.3		Desktop & Screen Saver	published	
8	2.3	2.3.2	Secure screen saver corners	published	Automated
9	2.4		Sharing	published	
10	2.4	2.4.10	Disable Content Caching	published	Automated
11	2.4	2.4.11	Disable Media Sharing	published	Automated
12	2.5		Security & Privacy	published	
13	2.5	2.5.3	Enable Location Services	published	Automated
14	2.5	2.5.4	Monitor Location Services Access	published	Manual
15	2.5	2.5.5	Disable sending diagnostic and usage data to Apple	published	Automated
16	2.5	2.5.7	Camera Privacy and Confidentiality Concerns	published	Manual
17	2.5.1		Encryption	published	
18	2.5.2		Firewall	published	
19	2.6		iCloud	published	
20	2.6	2.6.1	iCloud configuration	published	Manual
21	2.6	2.6.2	iCloud keychain	published	Manual
22	2.6	2.6.3	iCloud Drive	published	Manual
23	2.6	2.6.4	iCloud Drive Document and Desktop sync	published	Manual
24	2.7		Time Machine	published	
25	2.7	2.7.1	Time Machine Auto-Backup	published	Automated
26	3		Logging and Auditing	published	
27	3	3.2	Configure Security Auditing Flags per local organizational requirements	published	Manual
28	3	3.7	Software Inventory Considerations	published	Manual
29	4		Network Configurations	published	
30	4	4.1	Disable Bonjour advertising service	published	Automated
31	4	4.3	Create network specific locations	published	Manual
32	4	4.8	Review Wi-Fi Settings	published	Manual
33	5		System Access, Authentication and Authorization	published	
34	5.1		File System Permissions and Access Controls	published	
35	5.1	5.1.4	Check Library folder for world writable files	published	Automated
36	5.2		Password Management	published	
37	5.2	5.2.3	Complex passwords must contain an Alphabetic Character	published	Manual
38	5.2	5.2.4	Complex passwords must contain a Numeric Character	published	Manual
39	5.2	5.2.5	Complex passwords must contain a Special Character	published	Manual
40	5.2	5.2.6	Complex passwords must uppercase and lowercase letters	published	Manual
41	5	5.4	Automatically lock the login keychain for inactivity	published	Manual
42	5	5.8	Ensure login keychain is locked when the computer sleeps	published	Manual
43	5	5.10	Ensure system is set to hibernate	published	Automated
44	5	5.14	Create a Login window banner	published	Automated
45	5	5.16	Disable Fast User Switching	published	Manual
46	5	5.17	Secure individual keychains and home	published	Manual
47	6		User Accounts and Environment	published	
48	6.1		Accounts Preferences Action Items	published	
49	7		Appendix: Additional Considerations	published	
50	7	7.1	Extensible Firmware Interface (EFI) password	published	Manual
51	7	7.3	FileVault and Local Account Password Reset using AppleID	published	Manual
52	7	7.4	App Store Password Settings	published	Manual
53	7	7.6	System information backup to remote computers	published	Manual
66	5.2	5.2.7	Password Age	published	Automated
67	5.2	5.2.8	Password History	published	Automated
68	5	5.3	Reduce the sudo timeout period	published	Automated
69	5	5.5	Use a separate timestamp for each user/tty combo	published	Automated
70	5	5.7	Do not enable the "root" account	published	Automated
71	5	5.8	Disable automatic login	published	Automated
72	5	5.9	Require a password to wake the computer from sleep or screen saver	published	Manual
73	5	5.11	Require an administrator password to access system-wide preferences	published	Automated
74	5	5.12	Ensure an administrator account cannot login to another user's active and locked session	published	Automated
75	5	5.13	Create a custom message for the Login Screen	published	Automated
76	5	5.15	Do not enter a password-related hint	published	Automated
77	5	5.18	System Integrity Protection status	published	Automated
78	5	5.19	Enable Library Validation	published	Automated
79	6		User Accounts and Environment	published	
80	6.1		Accounts Preferences Action Items	published	
81	6.1	6.1.1	Display login window as name and password	published	Automated
82	6.1	6.1.2	Disable "Show password hints"	published	Automated
83	6.1	6.1.3	Disable guest account login	published	Automated
84	6.1	6.1.4	Disable "Allow guests to connect to shared folders"	published	Automated
85	6.1	6.1.5	Remove Guest home folder	published	Automated
86	6	6.2	Turn on filename extensions	published	Automated
87	6	6.3	Disable the automatic run of safe files in Safari	published	Manual
88	7		Appendix: Additional Considerations	published	
89	7	7.3	Repairing permissions is no longer needed	published	Manual
90	7	7.5	Apple Watch features with macOS	published	Manual
91	7	7.7	Unified logging	published	Manual
92	7	7.8	AirDrop security considerations	published	Manual
93	7	7.9	Touch ID	published	Manual
94	7	7.10	Sidecar	published	Manual
95	7	7.11	Screen Time	published	Manual

In case you're not aware, this is a list of what controls get put in place on an endpoint when you deploy CIS level 1.
We required level 2 which has more…but this also doesn't consider of the software needed for compliance
We have to test and deploy things like cmdReporter (now Jamf Compliance reporter), Anti-virus, firewall, and this is not even considering OS updates and software patching which is also part of security enforcement
So yeah, pushing a button was out of the question….
What we decided to do was break down the controls using some metrics and deploy in waves!



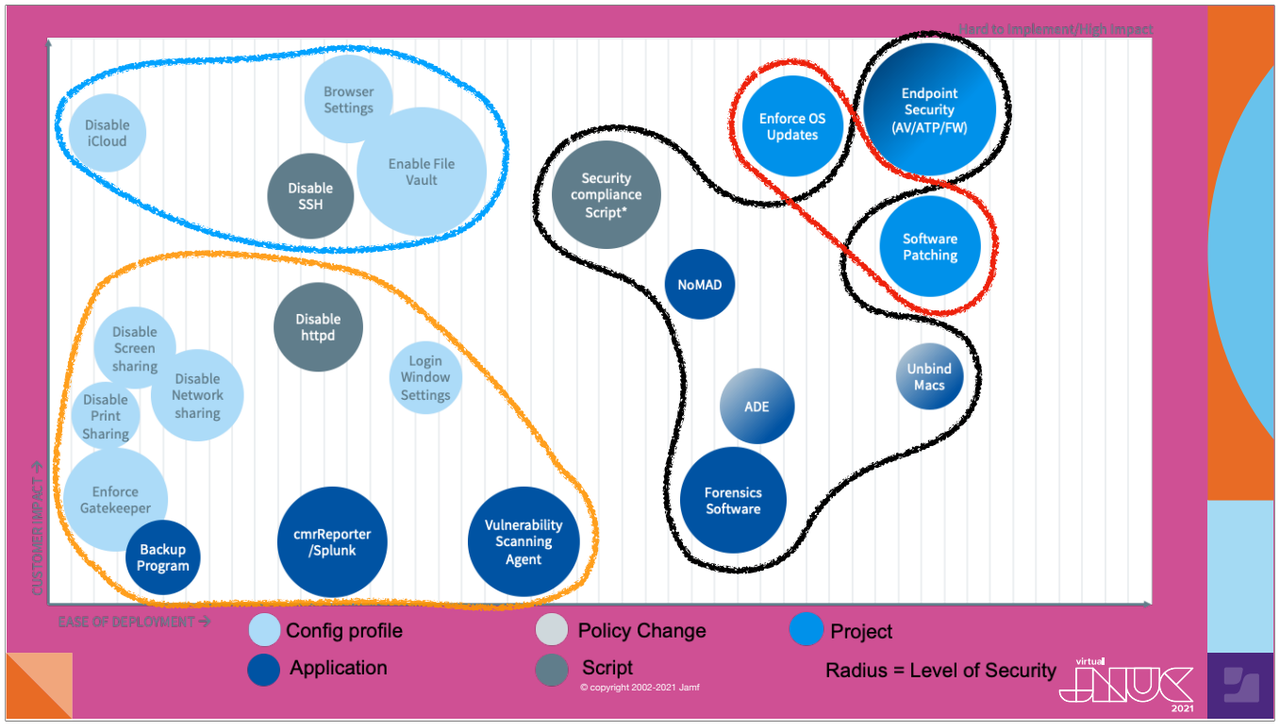
We defined three basic metrics:
In order of importance in our decisions: User impact, Ease of deployment and level of security

How much is this going to affect the user? We weighted this heavily in the beginning. We decided to basically deploy all settings that didn't affect the user a lot in the beginning.

How easy is the config to deploy? Config profiles are easy to make and deploy, scripts are harder. AV installs are simple but tuning them is really difficult and time consuming..so maybe not a good thing to do first. We also decided to say "if these three settings are in the same config profile…let's just try to add the rest that are needed"

And last, how much more secure is the device after implementation. Deploying a config to show a Login window message doesn't make the device much more secure, but enforcing filevault does.

So now, with the metrics defined, we took a look at the settings we had to deploy and started "scoring". Keep in mind, we didn't have a number value associated with each of these, it was more of an ad hoc scoring system but I'm sure you can probably define some scores for these if you really needed.



We took our metrics, and we started plotting things out…
Now, this isn't every check…it's not even a quarter of the checks, but this was the idea.

You can see the legend there…

We grouped things as best we could based on where they fell in the chart here. So :click: wave 1 here was all the simple, unobtrusive settings…lots of config profiles and silent agents that usually don't affect the user. You can see cmdReporter (now Jamf Compliance reporter) and Splunk, our enterprise backup solution, but also disabling of all of the sharing macOS has.

:Click: Wave 2 was more configs and some scripts…these were more user facing changes and things that users may be upset with…disabling icloud, changing browser settings, things like that

Finally, the third wave are things that take time to do and will be very noticeable to the user. Our AV, unbinding the Macs so we can use automated enrollment…and wave 3 was also used for cleanup and adding anything we removed from previous waves based on user feedback.

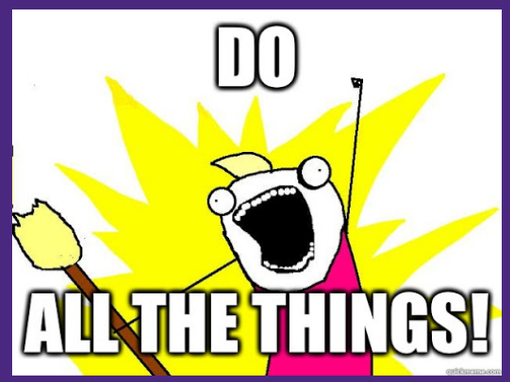
Now…:click: there are 2 here that are marked as "Project." Enforce OS updated and software patching. We basically broke these down into their own sub-projects with their own schedules. We also knew that these would continue after the main project to secure the environment. These were a mix of "secure and enhance". When I go over lessons learned, you'll see why this was very important.

Now that we knew how many waves we needed..we made our timeline and we were ready to start communicating things out!

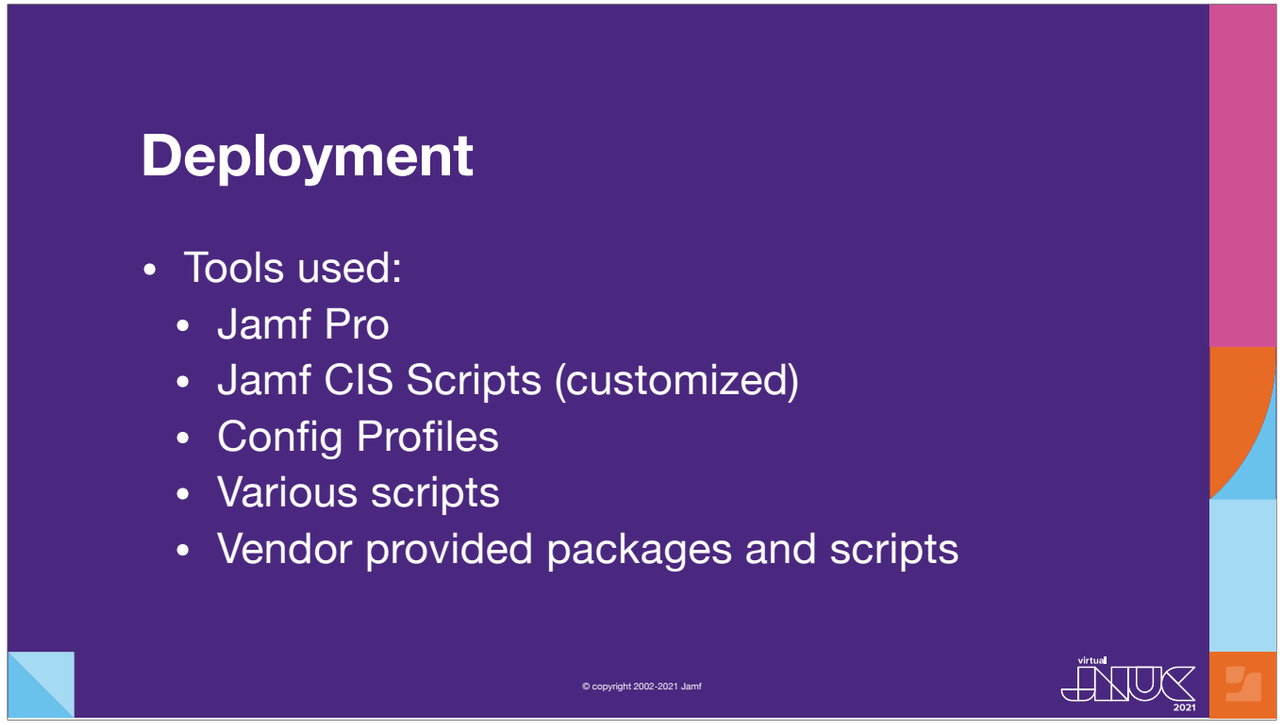
A good communication plan is a requirement for successful deployment. Repeat.
A few pieces of advice with regards to creating an effective communication plan
When communicating, you want to let your users know about very noticeable changes and major policy changes…but you shouldn’t let them know every little thing
You should let users know about changes like a login window privacy policy or installing a new application (even if it’s not right in their face)
You rally want to let them know about things like screen lock time and of course AV installation…these are things they care about, and they’ll notice.
You don’t need to tell them you’re installing a log forwarder or adding a shell MOTD
Your list may vary of course
You want to send out the comms with enough time for users to take in the info and provide feedback, remember…even bad feedback is feedback, and you should take it all into consideration
An example here: we communicated out that we were disabling screen sharing using VNC in wave 1, we received feedback that lots of people used this functionality for collaboration…we gave the users advice on new solutions and we told them we’re moving it to wave 2. The users were happy, we were happy.
And one very important one that often gets overlooked: give positive impacts of the changes
Yes, you’re taking some things away and altering settings of machines but there are many positives to this
“We are pushing an updated Office client to your machine so that you can have all of the most updated features and to ensure you can get support when needed”
“We’re deploying NoMAD to your machine so that your local password can synchronize with your network password, you only need one password!”
“The backup solution will be deployed to ensure your data is safely backed up in the event of a problem.”
Remember, we’re trying to rebuild relatinships…and this is one good way to do that.

Implementation

Do all of the things…..IN WAVES!



And finally, it’s time to implement the changes. You panned and planned, and you’re ready for wave 1!



We used our Jamf pro server to deploy everything here
The Jamf CIS scripts worked very well for us but we customized a bunch of checks and also made different remediations based on each wave.
The config profiles were a mixture of using the built in Jamf UI and making our own with Profile Creator or the later tool of iMazing Profile Editor, cannot recommend them enough..make sure you sign your profiles before uploading to jamf
You will need to make custom profiles…jamf has updated the UI to allow much easier custom application config profiles, use them where possible
We used various scripts found on Git provided by the community and we also found we had to write out own to do some things
Some of our security vendors provided some packages, config profiles, and scripts
Make sure you look at these closely before deployment, I use suspicious package by mothersRuin to look at every vendor supplied package…I cannot tell you how many times I've had to send packages back to them because they had issues

So…

Who goes first?

So you have your tools set up..now..who goes first?

…

Early adopters!

Early Adopters

- Find them **early**
- Get a good mixture of user types
  - IT staff make great early adopters
- Communicate and follow-up

virtual JNUC 2021

This is important..when I was demoing this presentation to my partner she told me to make this really stand out…so I gave it it's own slide

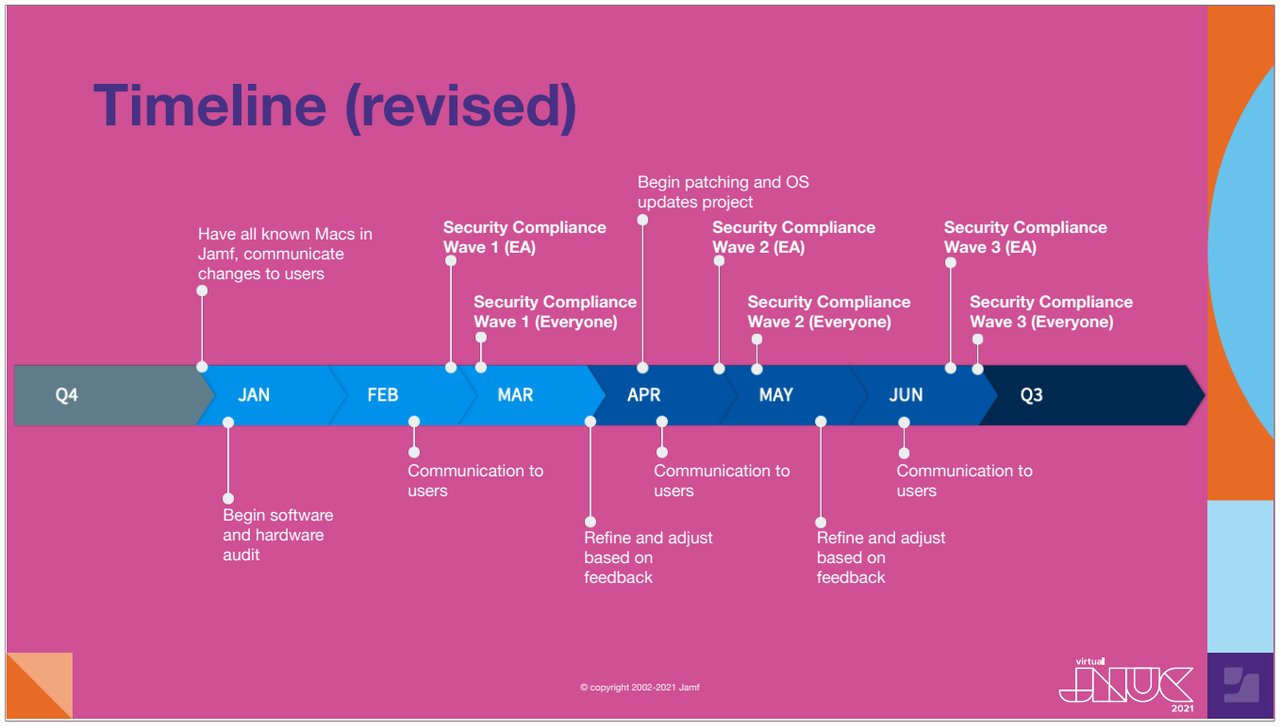
Find your early adopters early in the process. This should go in the lessons learned, but I'm making sure that you know this…find your early adopters early.

Find a good mixture of your user types…if you have a lot of devs and a few art directors, get a handful of devs and one art director to be an early adopter. Take a good percentage of your environment and directly email them and ask them if they can be an early adopter. We have 17 early adopters…but we also have 17 users in the IT department that get things early. That's another good built-in testing group…IT.

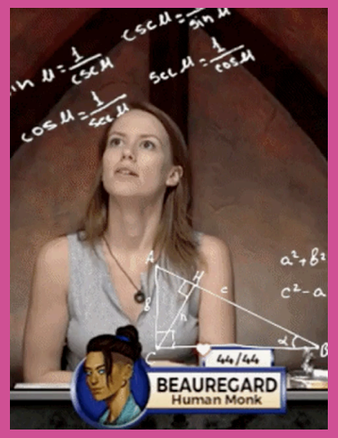
Whomever you choose, communicate and follow-up….make sure they know they'll be be getting changes before everyone else..maybe a week or 2. Make sure they know things may break…but also make sure they have a direct line to you to immediately fix something if you break it.

Always follow-up with them…ask for feedback, interact with them. These users are your last line of defense from pushing out a work stopping change.

They can save your users from disaster and save you from losing your job.



Now that we knew how many waves we needed..we made our timeline and we were ready to start communicating things out!



$48,000
Under budget

1
Data loss incident

75%
Decrease in vulnerabilities per Mac

Okay, the end of the project….by the numbers. Our rollout was a rousing success!

We completed the project on time and under budget by about $48000. We completed wave 2 much late than planned but wave 3 was completed just before the deadline…so we called that a success overall.

We did have 1 Data loss incident…now we don’t know if this was user error or caused by the changes, but we found it to be important enough to report on

And thanks for Nessus scans and patching we saw a 70% decrease in vulnerabilities in the environment. More on that…

Total Devices	Low	Medium	High	Critical	Total	Ratio (avg Vulns per Device)
260	64	291	1131	243	1729	6.38
254	65	232	888	358	1543	6.07
259	62	219	1214	217	1712	6.61
223	63	905	2944	493	4405	19.75
280	[illegible]	929	3124	528	4830	17.25
284	333	1071	3116	528	5048	17.77
289	385	1055	3106	535	5081	17.58
291	375	632	1503	149	2659	9.14
302	390	285	1290	118	2083	6.90
302	393	229	1116	116	1854	6.14
299	382	176	1010	81	1649	5.52
288	407	292	908	56	1663	5.77
293	421	420	737	258	1836	6.27
297	417	298	154	62	931	3.13
299	418	163	160	153	894	2.99
299	418	163	160	153	894	2.99
297	415	133	255	210	1013	3.41
305	422	105	248	83	858	2.81
306	386	106	388	124	1004	3.28
294	368	190	330	221	1109	3.77
289	366	69	250	73	758	2.62
287	360	67	223	61	711	2.48
287	360	67	223	61	711	2.48
174	[illegible]	34	122	142	363	2.09
136	51	31	146	42	270	1.99
119	35	25	128	65	253	2.13
149	9	18	183	37	247	1.66
264	9	126	153	293	581	2.20
253	9	[illegible]	148	97	596	2.36
331	4	676	836	66	1582	4.78
331	4	676	836	66	1582	4.78
334	2	588	786	37	1413	4.23
341	2	596	750	105	1453	5.22
351	2	756	592	60	1410	4.02
346	2	771	551	110	1434	4.14
366	2	766	545	33	1346	3.68
371	20	654	560	116	1350	3.64
383	29	721	742	385	1877	4.90
387	30	848	809	84	1771	4.58
382	31	839	722	41	1633	4.27
392	31	913	690	31	1665	4.25

These are our actual vulnerability scan results… we went from a high of 19.75 per endpoint when we only had 223 devices….to an average of 4.25 in 392 devices.

A note about some of these…since there are some weird patterns here I want to explain.

:Click: Here, we found a vulnerability with a piece of security software…it was patched in December.

And here…:click: we see the Apache vulnerability in 10.14 and 10.15. Apple patched it the week after listed…so some of those numbers are a bit off

Keep in mind, some of these are also right after an OS update was released or a Chrome zero day was found..but still overall, good numbers!

Lessons Learned

- Identify early adopters first
- Focus on patching separately
- Write more KBs and user support docs
- Find support people to help
- It’s better to be late than wrong
- Don’t ignore Macs!

So what did we learn throughout the project? What could we have done to make things better?

Find your test group early…basically find them right as you’re enrolling devices…if you already have devices enrolled…find them now
As I mentioned before, patching ended up being a separate project bt we didn’t realize it until a few months in. We should have identified someone to tackle packaging software updates and packages instead o doing it ourselves while also securing the platform. Yes, they’re part of the same goal…but patching takes a lot of time and you can very easily train someone how to package software and deploy it. If you’re lucky enough to have an operations team or person, use them and start early.
I didn’t have time to go into this today, but you should be providing KB articles and other user support docs with your communications…and if you think you have enough, write a few more. It’s never enough!
This is especially important when you’re introducing new apps and settings to both your users and the support staff
A good time to mention…take the time to identify support champions..people who express interest in supporting macs..this will help free you up from things like helping users with printers or basic troubleshooting
We focused a lot on schedule and because of this, we had a data loss incident. Again, we didn’t know if it was our fault, but we knew that we rushed some controls and changes..either way, we knew we could have hanged something when it happened..and we did.
That last bullet is more of a general lesson learned for big corporations…:click: Don’t ignore macs. Invest in them! If you have Macs already, the longer you ignore them, the harder and more expensive it becomes to fit them into your environment.

Thank you for listening!

virtual JNUC 2021

Resources

- Jamf CIS Scripts: https://github.com/jamf/CIS-for-macOS-Catalina-CP
- macOS Security Compliance Project: https://github.com/usnistgov/macos_security
- CMMC Website: https://www.acq.osd.mil/cmmc/
- NIST 800-171 Rev. 2 Documentation: https://csrc.nist.gov/publications/detail/sp/800-171/rev-2/final