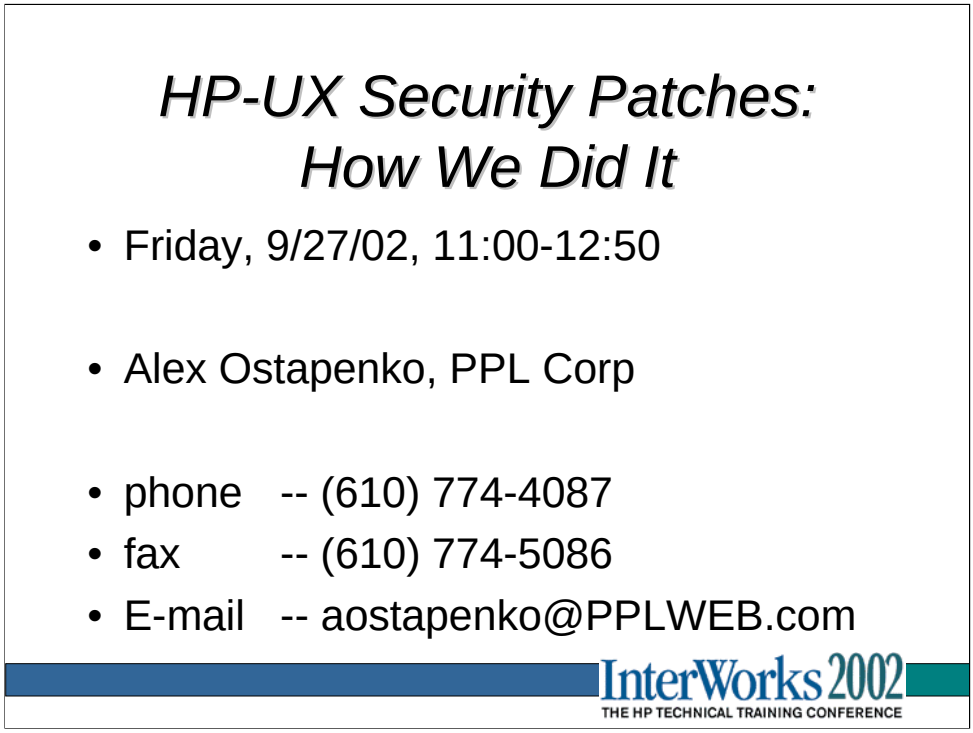
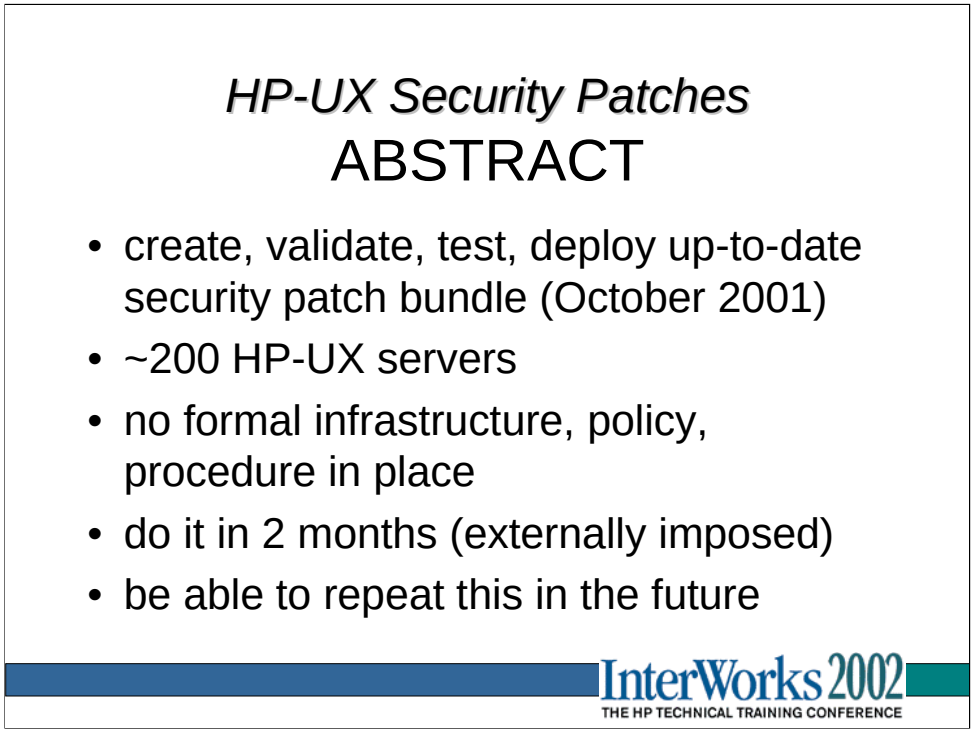


Biographical Sketch

Alex has been involved professionally in computers since 1977 over wide-ranging platforms such as the CDC-6400, DEC-20, IBM-370 architecture. He entered the UNIX world academically in 1989 while at night school at Villanova University, and leveraged that experience professionally starting in 1994. Alex has worked at PPL Corp since 1982.



- The team on which I participated was charged with creating, validating, testing, and deploying an up-to-date security patch bundle on a wide ranging enterprise of about 200 HP-UX servers.

- The server models ranged from E55's and HP715's to J2240's and V2250's.

- The OS versions ranged from HP-UX 10.10 through 11.00.

- The endeavor presented several different challenges.

1. We were proceeding down an effort that had not been attempted before here, and we had no policy or procedure to do so. What that meant was that everything was done from scratch.

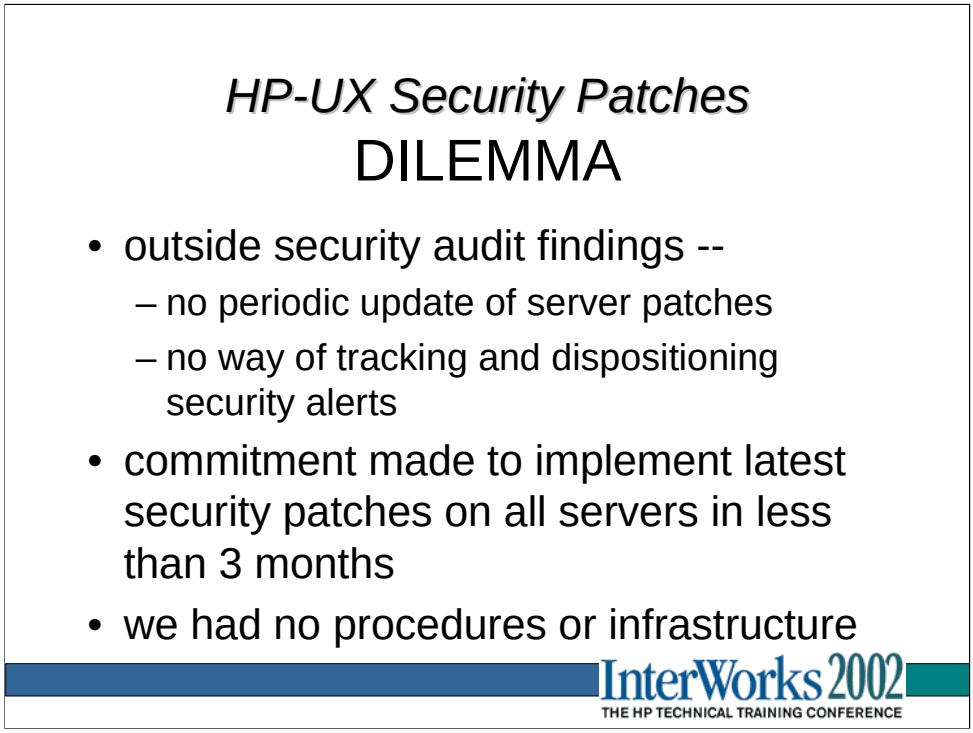
2. There was an urgency to complete the effort in a short period of time.

3. The project initially started with very limited resources, two part-time people.

4. An adequate testing facility did not exist, and there was great concern about the potential risks of rushing patches into production.

5. ITRC tools were cumbersome to use, and purchase of HP CSS was deemed too expensive.

In the end, the process took longer than expected, but was successful. The outcome was that a patching strategy and process project team was formed with the responsibility of creating a more proactive approach and procedure for security patching, and patching in general.



DILEMMA

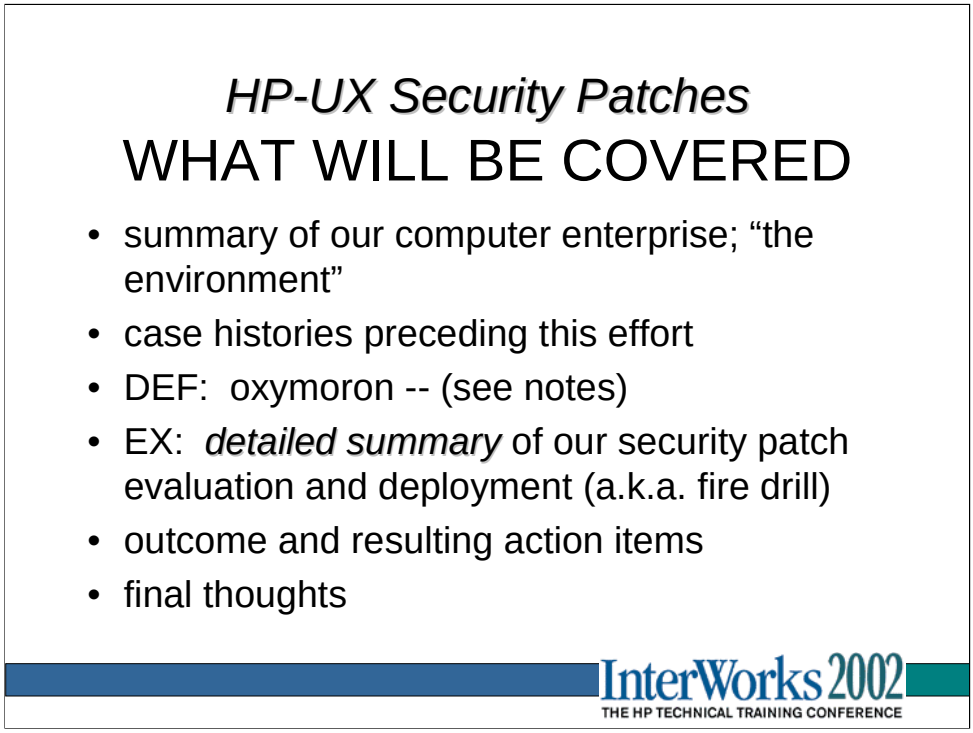
1. An outside security audit had a number of findings, the two that relate to this talk being --

(a) no procedure in place for periodic updating of patches

(b) no infrastructure to track, disposition, and if necessary implement alerts and their respective patches/fixes related to security

2. Out of the blue, a commitment to senior management was made that all current security patches would be deployed acrosw our whole environment in a little over two months.

3. There were no procedures or technical/people resource infrastructure in place to support an effort of this magnitude, even if we had more time. Best guess was a minimum of 6 months with present resources, that's what our Y2K project.



summary of our computer enterprise; "the environment"

It is helpful to understand the environment upon which the security patches were deployed. This includes the hardware platforms, the OS versions, the current standards, and other pertinent details.

case histories preceding this effort

Three case histories preceding the security patch effort will be described. In recounting the events, I will describe a number of the tools, shortcomings, problems, and challenges that shaped our thinking prior to the security patch deployment.

DEF: oxymoron

1. a rhetorical figure in which incongruous or contradictory terms are combined
2. figure in which an epithet of a contrary signification is added to a word

***detailed summary* of our security patch evaluation and deployment**

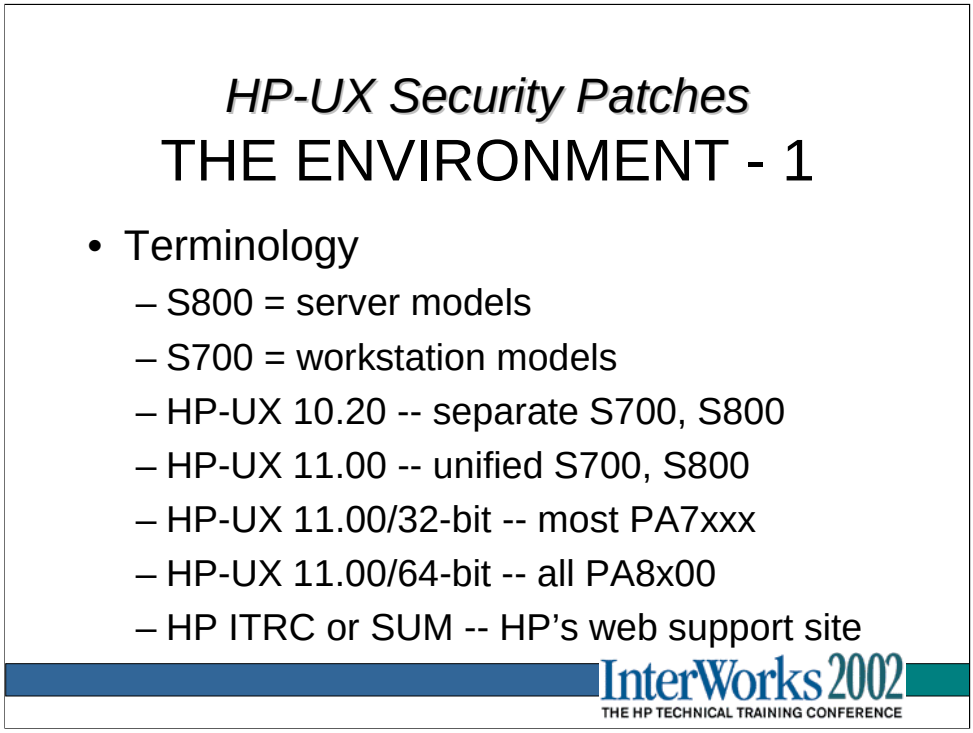
The actual events will show the many different facets of coordinating, evaluating, validating and testing, recovering, deploying, tracking, and mitigating risk of an effort of this magnitude.

outcome and resulting action items

From this particular event, a new project was begun to define, document, and implement an infrastructure for the risk-mitigated periodic deployment of regular OS patches.

final thoughts

Doesn't every presenter have final thoughts?



Note -- although HP's web site has gone through a number of changes since 1997, we will refer to it as HP ITRC throughout this presentation, and we will show examples using presently available tools even if the example itself pre-dates the availability of the tools.

HP-UX Security Patches

THE ENVIRONMENT - 2

- Hardware & OS (see “Notes” for details)
  - V22xx
  - N-class, L-class
  - K-class -- take your pick
  - D-class
  - alphabet soup
  - workstation class (a.k.a. S700)
  - other mfg’s -- Sun, IBM, NT

InterWorks 2002
THE HP TECHNICAL TRAINING CONFERENCE

V22xx

- 5 servers at HP-UX 11.00/64-bit
- 5 consoles at HP-UX 10.20

N-class, L-class

- dozens and dozens (like McDonald’s)
- HP-UX 11.00/64-bit

K-class -- take your pick

- two K400’s originally at 10.00 and 10.01
- K420, K460, K570, K580
- all flavors of HP-UX from 10.20-->11.00

D-class

- D310, D320, D350, D370 (lots of these)

alphabet soup

- E35, E55, G40, G60, H60, H70, I70

workstation class (a.k.a. S700)

- HP735, HP712, HP715
- J2240
- HP-UX 10.20, 11.00/32-bit

other mfg’s -- Sun, IBM, NT

HP-UX Security Patches

THE ENVIRONMENT - 3

- Storage
  - internal disks and JBOD's (Jamaica boxes)
  - EMC
  - Clariion
  - FWD SCSI, Fibre, SAN

THE ENVIRONMENT - 4

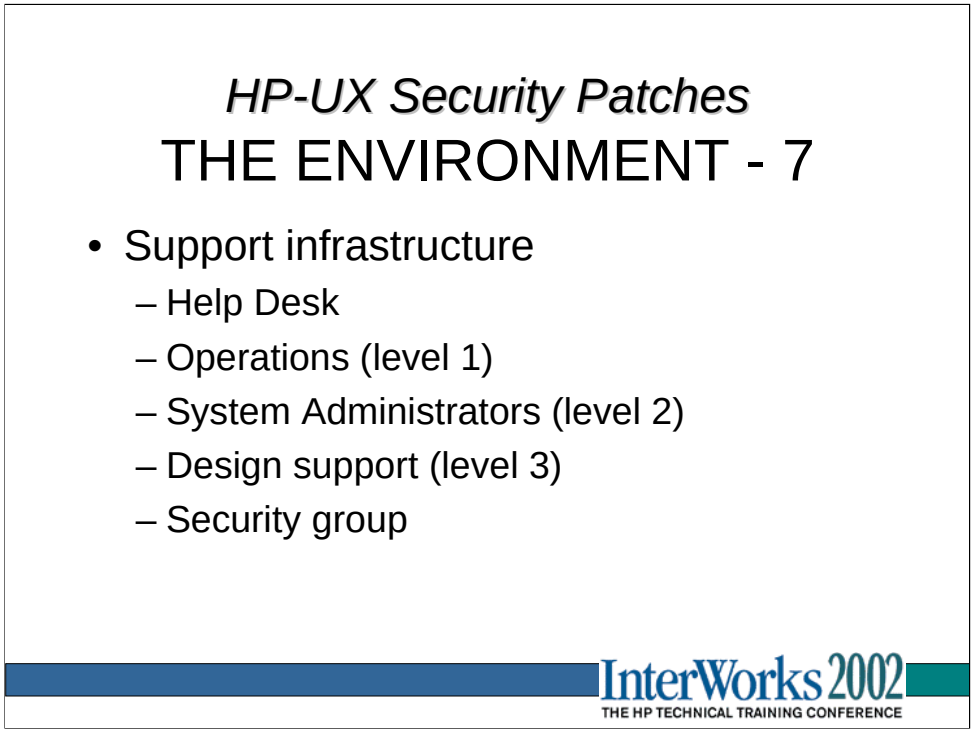
- Software
  - HP ANSI/C
  - Mirrordisk (Advanced-JFS)
  - GlancePlusPak (glance plus MWA)
  - Microfocus Cobol
  - Oracle DB (7.x-->8.1.7)
  - Connect:Direct
  - others...

THE ENVIRONMENT - 5

- Network
  - mostly 100Mbit
  - occasional 10Mbit
  - ATM --> 1Gbit backbone
  - Internet connectivity
  - telnet, rlogin, Hummingbird Exceed, X-term
  - FTP, NFS, rcp, rdist

THE ENVIRONMENT - 6

- Test Lab
  - DO NOT have representative hardware
  - do have representative software, but...
  - DO NOT have representative applications
  - DO NOT have representative storage
  - DO NOT have representative network
  - do have representative remote accessibility



Help Desk

- call menu, create problem tickets
- forwards calls to next level (usually level 2)

Operations (level 1)

- monitoring of automated alerts at central console, page outs
- job scheduling administration

System Administrators (level 2)

- do most of the work, traditional sys admin support
- small cadre responsible responsible for large enterprise
- do not have single person responsible for specific server

Design support (level 3)

- create standards for OS installs and depots for standard patch installs
- order servers
- manage Test Lab
- performance analysis

Security group

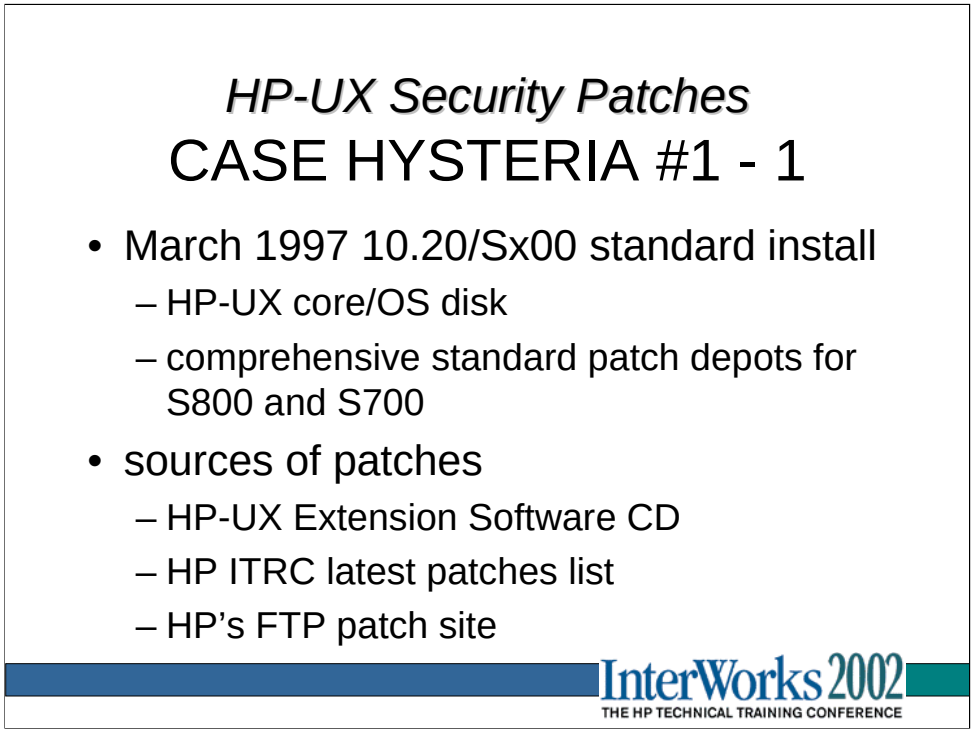
- report directly to CIO
- establish policies (a.k.a. write white papers)
- periodic announced/unannounced audits
- evaluating and recommending security measures
- tracking and dispositioning alerts
- administering computer access and root security

THE ENVIRONMENT - 8

- Change management philosophy
  - servers standardized as much as possible
  - change tracking memo system
  - 48 hour notice for changes
  - when required or desired, coordination with application owner in addition to the change memo

THE ENVIRONMENT - 9

- Server maintenance/outage constraints
  - certain applications have outage blackout periods, e.g., plant maintenance
  - certain applications can only go off-line at odd hours, e.g., Sunday 2-4am
  - certain applications require all their server to be at same OS and patch level



HP-UX Extension Software CD

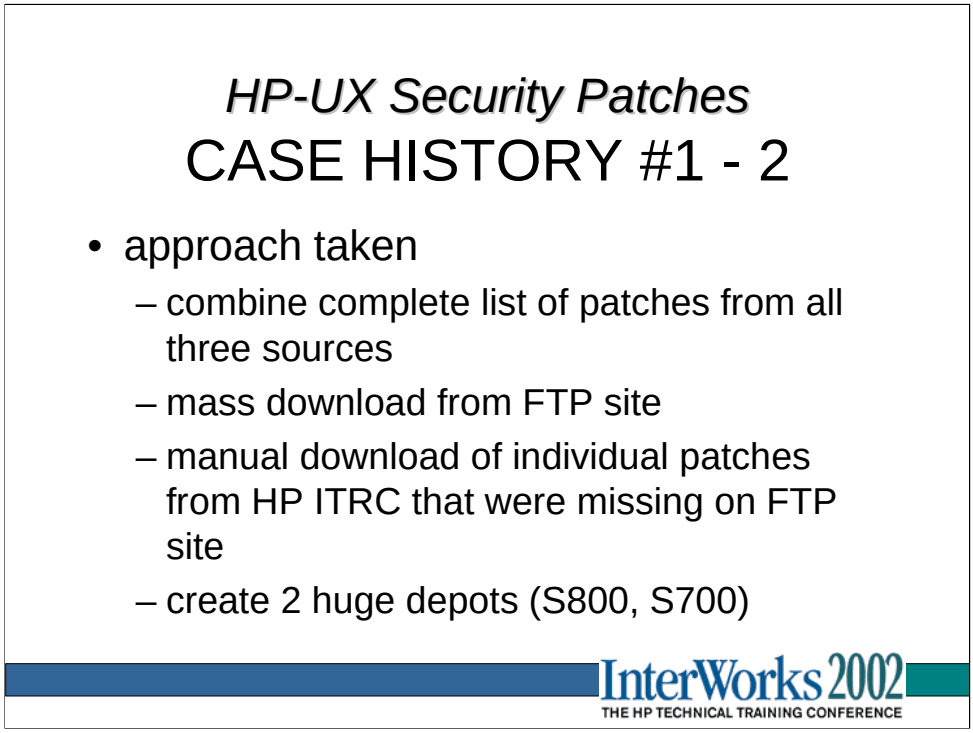
- for a particular vintage CD, patches are already 3 months old on it
- CD contains list of superceded or bad patches (if they knew the supercedes, why weren't they already included on the CD)
- the list in the previous bullet is oftentimes already superceded on HP ITRC
- CD release itself can potentially be bad; has occurred at least two times since 1997

HP ITRC list of patches

- did not seem complete, i.e., short list
- no easy method for package download
- not all patches listed on HP ITRC were available on HP's patch FTP site

HP patch FTP site

- cannot easily determine latest patch and dependencies
- missing patches from other sources



combine list of patches from all three sources

- obtain list of patches
  - •Extension software -- ls -l
  - •FTP site -- FTP ls
  - •HP ITRC -- save web page, script to parse HTML and extract patches
- combine lists

  cat cd.list ftp.list itrc.list | sort -u >combined.list

mass download

- script that FTP "gets" all the patches that are available on the FTP site

manual download

- grep -v ftp.list combined.list >not-found.list
- hop onto HP ITRC to get those patches in "not-found.list"

create large depot

- swcopy ????

HP-UX Security Patches

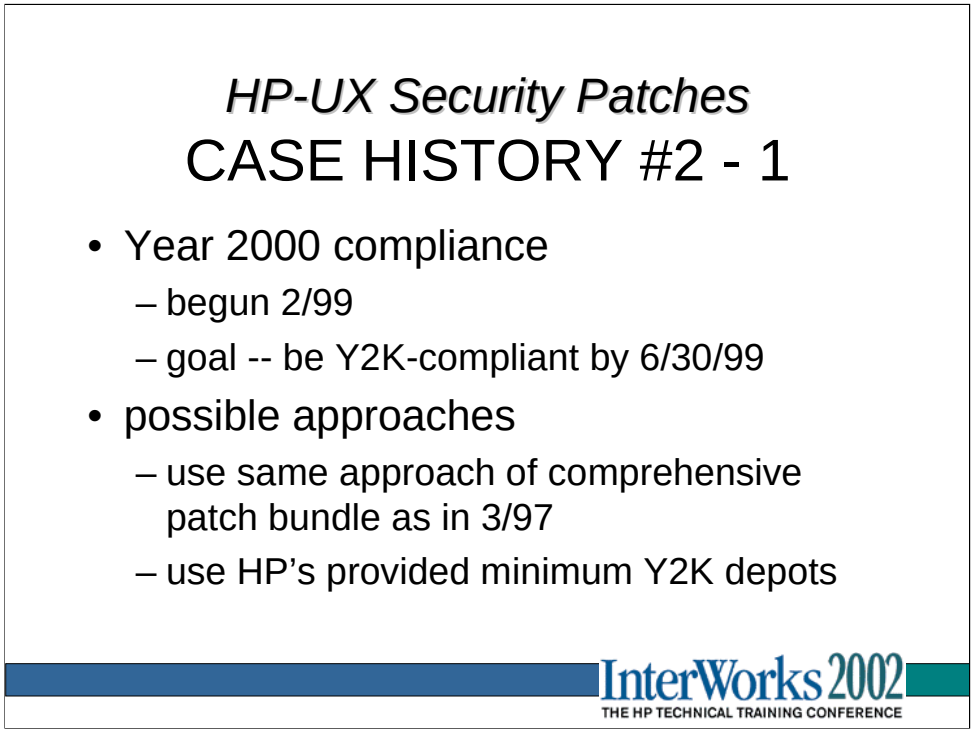
CASE HISTORY #1 - 3

- Shortcomings
  - time-consuming to create
  - patch texts and dependencies were not explicitly reviewed
  - took a long time to deploy to all servers
  - either did not or could not create an automated and repeatable procedure

HP-UX Security Patches

CASE HISTORY #1 - 4

- Advantages
  - only two depots -- S800, S700
  - every new server got a standard install regardless of its level of Instant Ignition
  - every old server got up-to-date patches
  - consistency across the enterprise

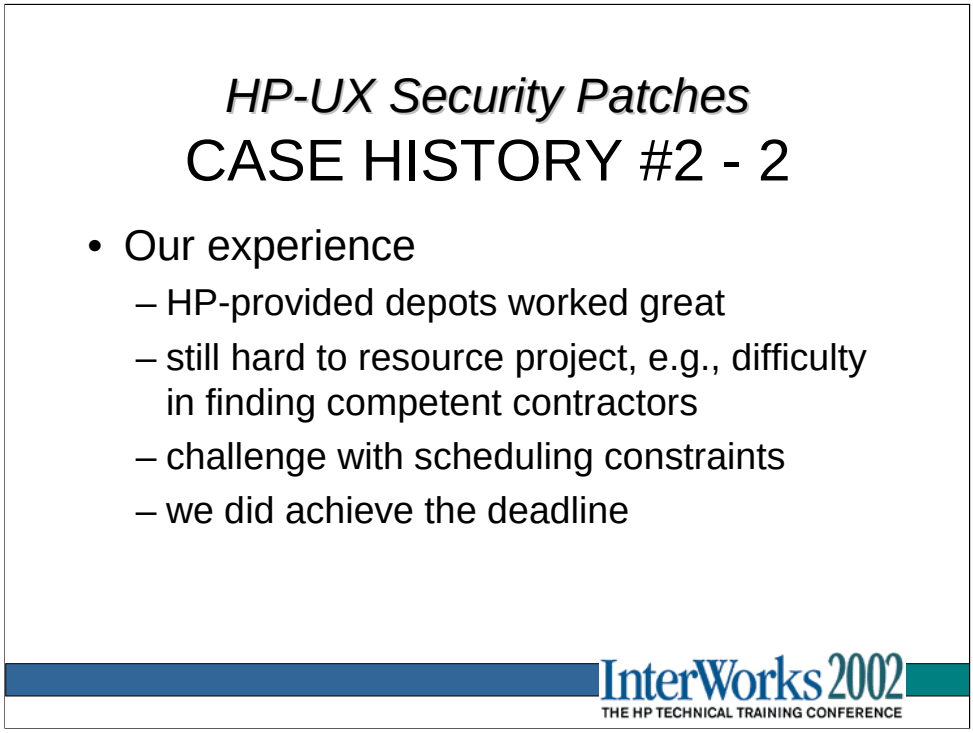


possible approach #1 -- comprehensive patch bundle

- appeared as if it might take too long because of the required time to create the new depots, the number of servers to be updated, and the scheduling constraints
- we did not have enough people to support a focused fast-track effort of this magnitude
- we were worried about not having adequate time or facility to validate the new large patch depots.. might we introduce new problems?

possible approach #2 -- minimum HP Y2K patch depot

- depot already certified by HP
- minimum number of patches, therefore, less opportunity for problems with patch install
- considerable information on HP ITRC supporting the Y2K patch depots and required steps to achieve compliance
- might be easier to resource this approach



HP-provided depots worked great

- did not have to worry about bad patches or missing dependencies
- could rely on patches being well-tried
- had certification for Y2K-compliance

still hard to resource project

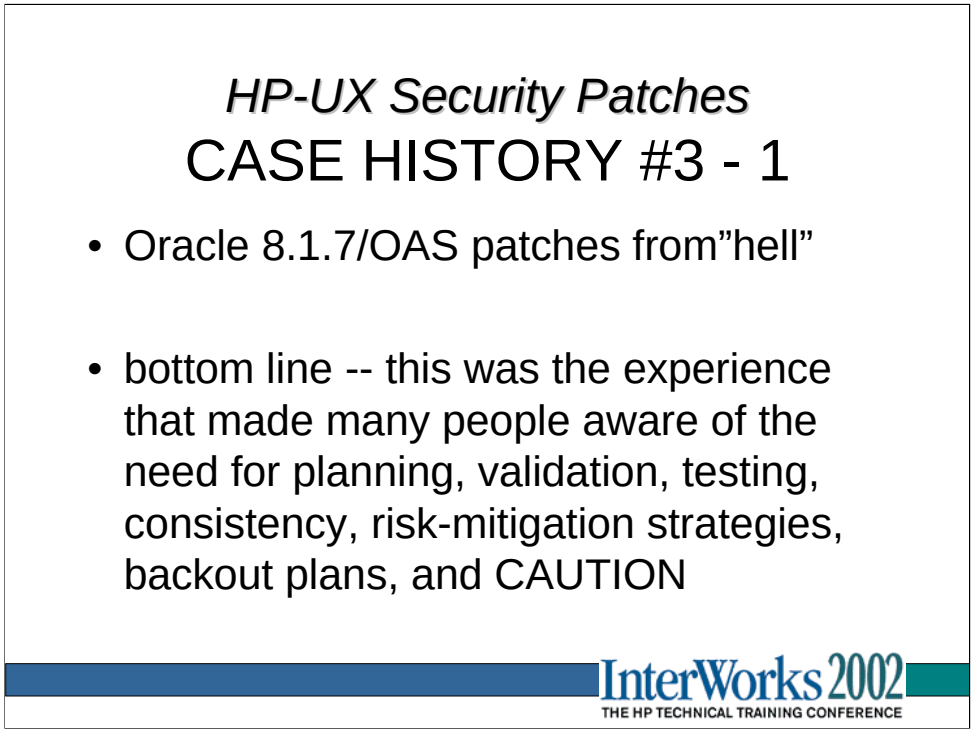
- e.g., difficulty in finding competent contractors
- went through nearly half-dozen contractors
- those competent were not willing to be team-players, i.e., they came from the old school of sys admin being Lord of the Server
- example -- had problems with one server not having enough space in /var for /var/adm/sw; instead of reporting back to seek advice for next step, completely redesigned disk layout to build larger filesystem; he ended up breaking applications; we ended up breaking his contract
- we ended up resourcing ourselves without any contractors; key projects

challenge with scheduling constraints

- those mentioned previously still applied here
- scrambled to complete by the deadline

we did achieve the deadline

- went on to add the October/November 1999 Y2K-update patch depot
- had no problems with Y2K



planning -- a proposed course of action or procedure worked out beforehand for the accomplishment of an objective

validation -- to establish the soundness of

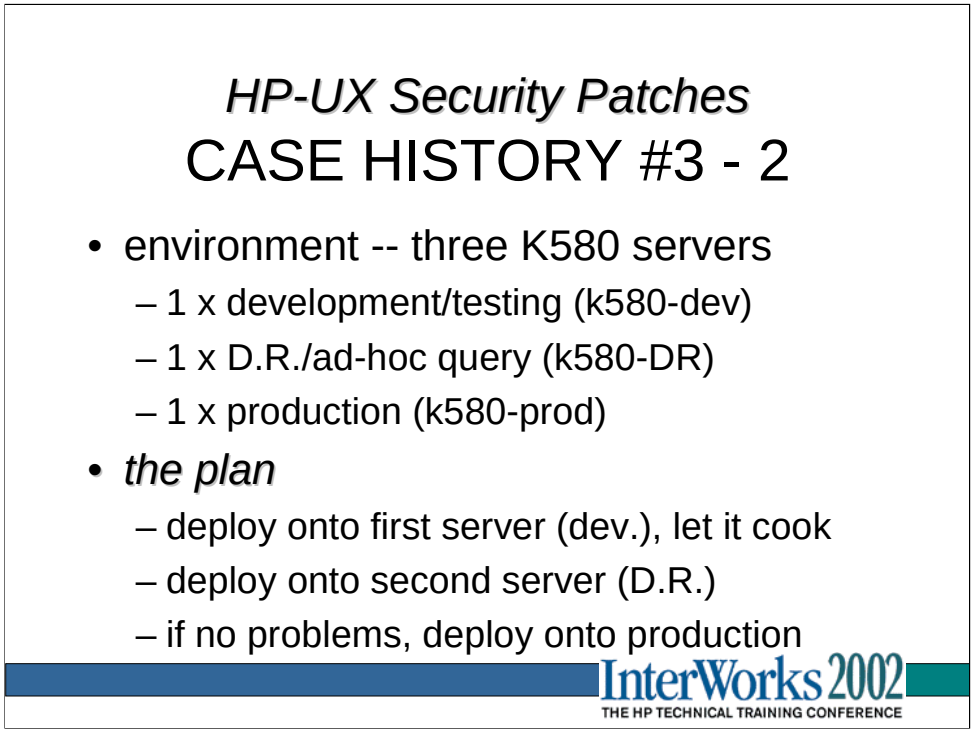
testing -- make sure the patches and depot install procedures work prior to their deployment onto production servers

consistency -- to make sure all targeted servers have the same configuration

risk-mitigation strategies -- steps taken to reduce the likelihood of problems

backout plans -- procedures that allow one to revert back to a previously known working state

caution -- proceeding carefully, methodically, and deliberately

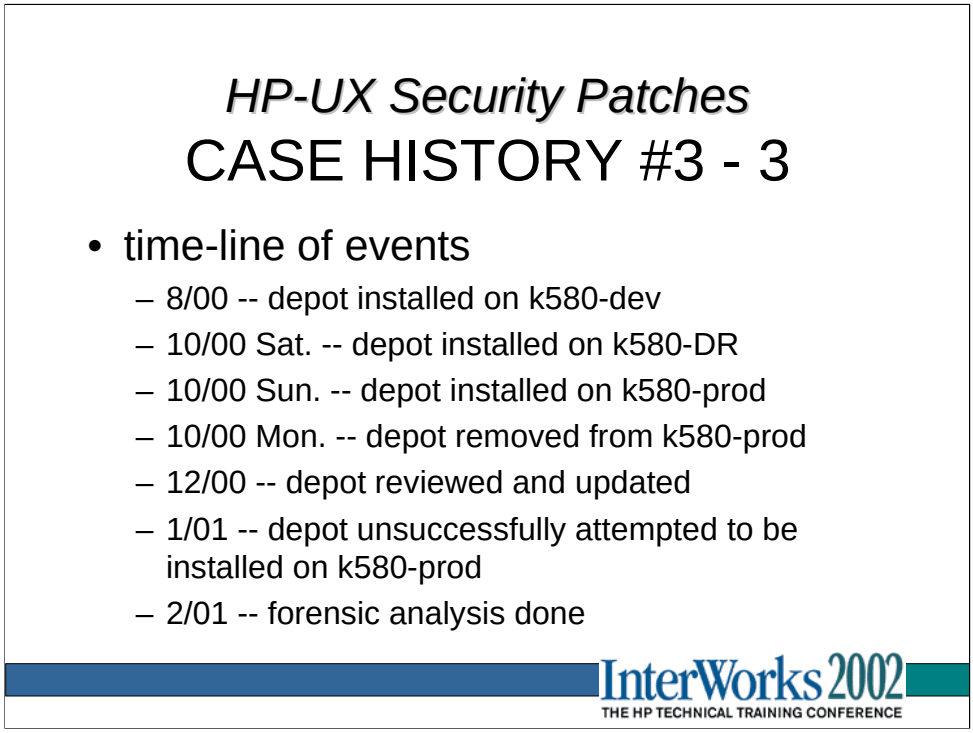


THE PLAN

1. Deploy the patches onto the development server, and let it cook for at least several weeks under typical development and test loads. In addition, execute some typical sys admin tasks, like changing kernel parameters, to make sure nothing was broken by the patches.

2. Deploy same patches onto the second server (D.R./ad-hoc query). This one is more representative of production. This would be done over a weekend. If successful, the production server would be done the same weekend.

3. If step #2 was successful, deploy the patches onto the third server (production). This has to be done the same weekend as the second server, because that is the D.R. server, and both have to be the same.



8/00 -- depot installed on k580-dev

•successfully installed with no errors

•kernel parameter change requiring kernel rebuild worked

•server cooked under various, but not production, loads

10/00 Sat. -- depot installed on k580-DR

•successfully installed with no errors

•however, kernel parameter change requiring kernel rebuild DID NOT work

10/00 Sun. -- depot installed on k580-prod

•despite errors with k580-DR, decision was made to proceed with install on k580-prod

•appeared to install successfully with no errors

•kernel parameter change requiring kernel rebuild was NOT done

10/00 Mon. -- depot removed from k580-prod

•when product load appeared on server, severe performance problems were encountered appearing to be from only 1 of 4 CPU's being scheduled

•depot was attempted to be removed, but one of the patches would not uninstall

•/stand was recovered from a backup

12/00 -- depot reanalyzed and updated

•rechecked patch supercedes and dependencies; updated depot

1/01 -- depot unsuccessfully attempted to be installed on k580-prod

•never got past kernel relink

HP-UX Security Patches
CASE HISTORY #3 - 4

- Forensic analysis
  - servers were not consistent
  - original depot was bad
  - 2 of 3 servers had a bad SAM patch
  - production server had the bad “ar” patch
  - questionable decision-making

InterWorks 2002
THE HP TECHNICAL TRAINING CONFERENCE

servers were not consistent

•different patch depots had been installed at different times in the past making the three servers different from each other

•but it was thought that they were all the same, so the inconsistent behavior was unexpected

•hence, when things worked properly on the first server, that was no guarantee they would work on the remaining two

original depot was bad

•depot included the “infamous” PHKL_18543 LITS patch, but not all of its dependencies

•depot contained a bad patch that caused the CPU performance problem

•other patches did not have all their dependencies

•depot contained some “old” patches that needed to be superceded

k580-DR and k580-prod servers had a bad SAM patch

•k580-dev had PHCO_17792, a good SAM patch

•but the others had PHCO_16576, a bad patch

•the bad patch is what caused the relink following a kernel parameter change to fail

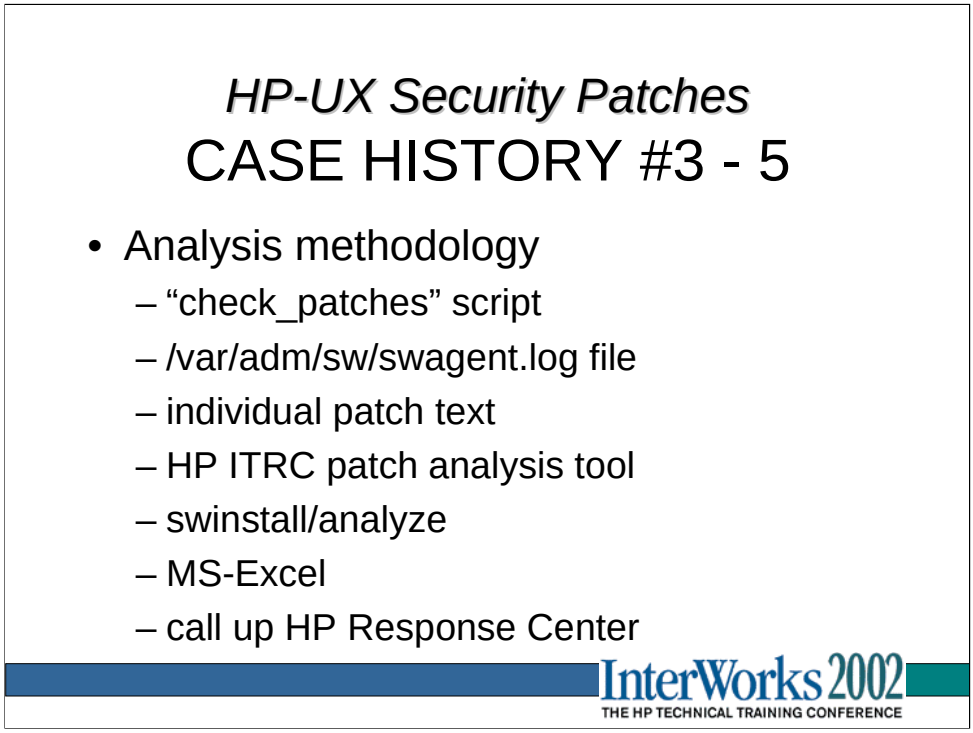
production server had the bad “ar” patch flagged by “check_patches”

•prevents correct updating of object modules causing the kernel relink to fail

•we didn’t have “check_patches” provided by patch PHCO_22044

questionable decision-making

•when error occurred with k580-DR, k580-prod should never have been updated that weekend



"check_patches" script

•contains a lot of good analysis

•flags some of the "dangerous" patches and incompletely or incorrectly installed patches

•installed by PHCO_24347 (formerly PHCO_22044)

/var/adm/sw/swagent.log file

•timeline of patch installs and removals

individual patch text

•oftentimes taken for granted

•can be seen on HP ITRC "individual patches"

•will contain warnings, dependencies, and special instructions

HP ITRC patch analysis tool

•helps determine conflicts, missing dependencies, and level of patch

•sometimes recommends newer versions of patches

•can use the configuration of a representative server to test proposed

swinstall/analyze

•pre-execution step of "swinstall" that checks for dependencies and conflicts

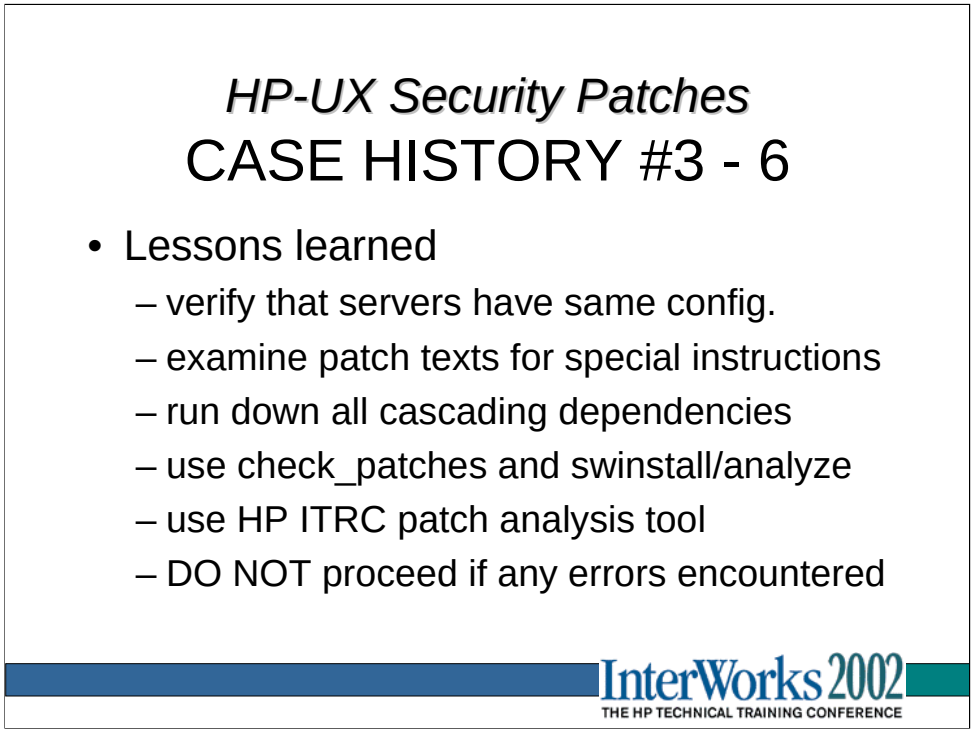
•can be done without actually installing patches

MS-Excel

•useful for tabularizing data such as installed/missing patches on servers, cascading dependencies, and level of patches

HP Response Center

•2nd-level support sometimes helpful in running down problem with a patch



verify that servers have the same configuration

•multiple servers supporting a particular application are expected to have the same configuration

•verify that fact

•you are depending upon this to validate testing on a less critical server prior to deploying on a more critical server

examine patch texts for special instructions

•special instructions alert one of potential conflicts (PHKL_18543), effects to applications (e.g., HP-OV), change in behavior (e.g., sendmail) and sometimes special extra dependencies not listed under “Dependencies”

run down all cascading dependencies

•you must check all patches listed under “Dependencies”

•you must then check the “Dependencies” section of the dependent patches

•and so on until all are accounted for

use check_patches and swinstall/analyze

•both commands used in advance of patch deployment may catch errors otherwise not seen

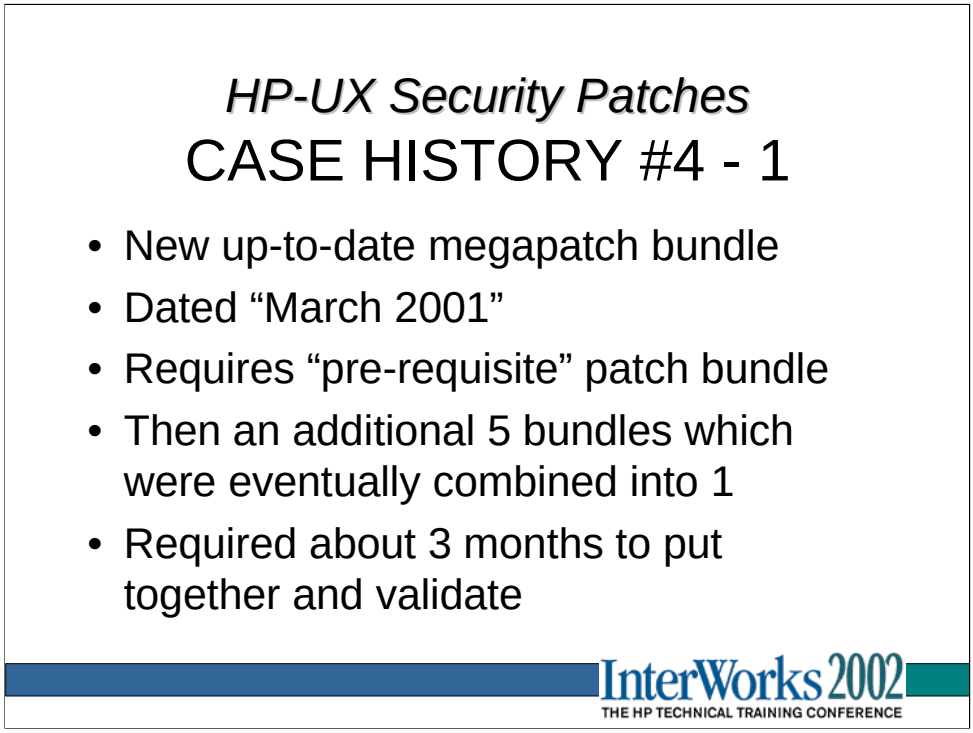
use HP ITRC patch analysis tool

•might resolve conflicts or missing dependencies

•might suggest newer versions of patches

DO NOT proceed if any errors are encountered

•if any deployment or testing steps don’t work, don’t proceed



New up-to-date megapatch bundle

•our new patch bundle standard

•to include an up-to-date set of patches to address the previous case history

•both for new servers and existing ones

•only for HP-UX 11.00; we did not worry about 10.20

Dated "March 2001"

•contained a smorgasborg of patches up to March

•needed to resolve warnings and conflicts

•creating one bundle for the enterprise, so it had to cover all our HW & SW

Requires "pre-requisite" patch bundle

•PHCO_21187 -- SAM patch

•PHCO_22044 -- "check_patches" and other tools

•PHCO_23966 -- updated Software Distributor

•PHSS_22514 -- updated "ld" and "linker" tools (the infamous "ar" patch)

Then an additional 5 bundles eventually combined into 1

•XSWGR1100(9/00)

•XSWCRHW1100(9/00)

•PHKL_18543 + dependencies

•Oracle 8.1.7 recommended + dependencies -- 14 patches ballooned to 30

•OAS recommended + dependencies

HP-UX Security Patches

CASE HISTORY #4 - 2

- Lessons learned
  - good planning and methodical approach can achieve success
  - task to create a “validated” depot is not trivial (took about 3 months part-time)
  - become familiar with the Software Distributor commands (swcopy, swinstall, etc.) -- see the “man” pages
  - be flexible: new warnings, problems, etc.

HP-UX Security Patches

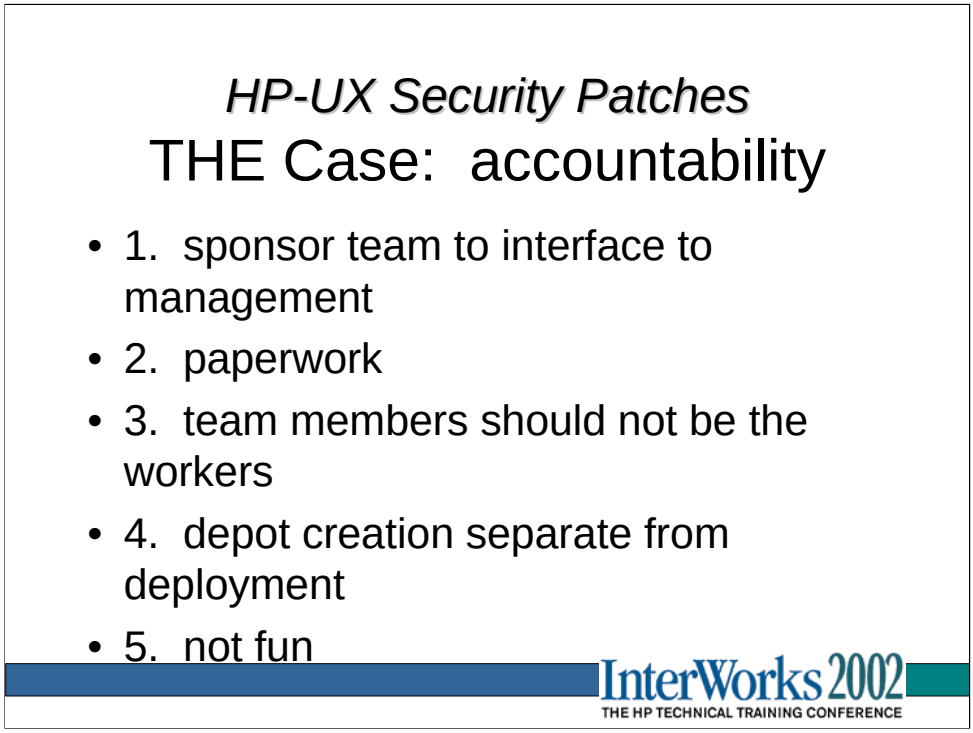
THE Case: the challenge

- At the end of October 2001, a commitment was made to have all enterprise servers (HP-UX, Solaris, NT) up-to-date with security patches by the end of December.
- Solaris patch tar-balls and NT SP's were a piece of cake
- Here we deal solely with HP-UX

HP-UX Security Patches

THE Case: What, us worry?

- accountability to management
- risk mitigation strategies
- external resources
- depot creation
- deployment and its coordination
- problems encountered
- the HP-UX 10.20 problem
- accomplishments
- follow-up actions



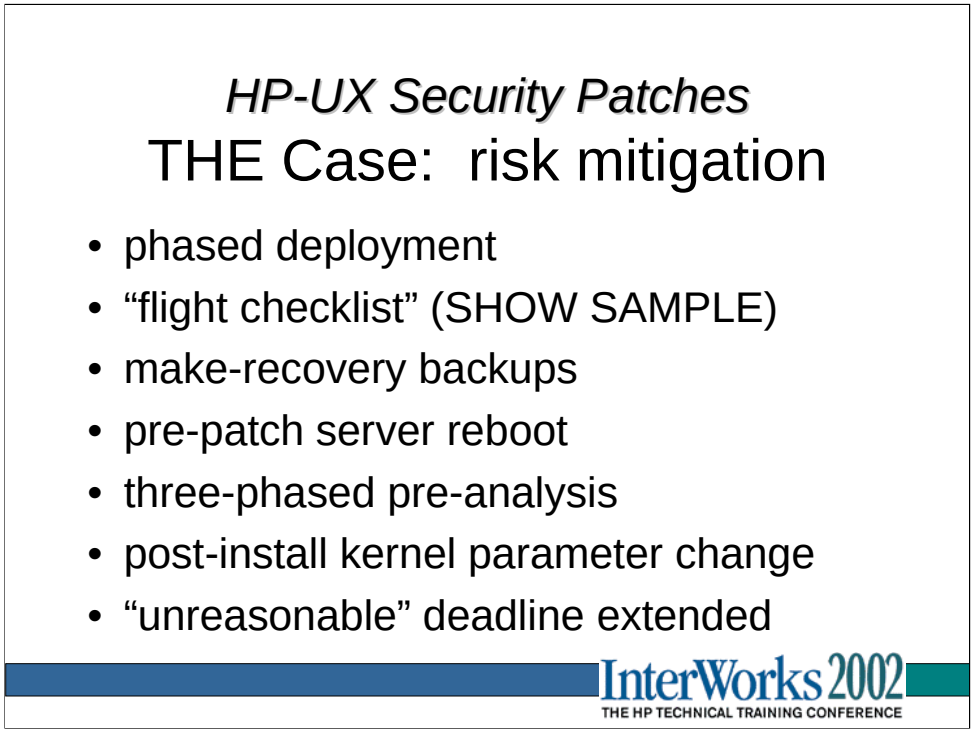
1. **sponsor team to interface to management** -- needed to track progress, authorize expenditures, resolve issues, report to management

2. **paperwork** -- there will be some involved (ha! ha!)

3. **team members should not be the workers** -- on a fast-track project such as this, workers did not have time for management meetings; try as they might, we succeeded on this point

4. **depot creation separate from deployment** -- cannot be the same people because after first depot is created, we would like deployment to start while the next depot is being created; in addition, our deployment team were the normal sys admins, and they had their regular work to do also

5. **not fun** -- meetings and interfacing to management never is; that's why I didn't spend too much time discussing this here



phased deployment

•test lab to production

•less critical to more critical applications

•within an application, test --> development --> DR --> production

•required at least 1 week cook time between phases

“flight checklist”

•need flaps and landing gear down when landing an airplane

•likewise, require following a checklist to assure all steps were properly completed

•easy to document problems or errors

2 x make-recovery backups

•on tape when device available or available to be attached

•otherwise, over the network

•required two copies because we already ran into problems in which 1 copy was bad

pre-patch server reboot

•to assure server was rebootable prior to applying patches

•only if server had not been rebooted in more than 90 days

three-phased pre-analysis (SHOW SAMPLE OF SCRIPT AND REPORT)

•homegrown script (get-swinfo) -- quickly obtains and reports on key patch configurations (such as PHKL_18543 and Ignite/UX) and summary of patching history extracted from /var/adm/sw/swagent.log

•check_patches script -- uncover any unexpected problems or bad patch installs

•swinstall/analysis -- find any errors and space problems in advance of applying patches

post-install kernel parameter change

•make sure that newly patched system can successfully rebuild the kernel

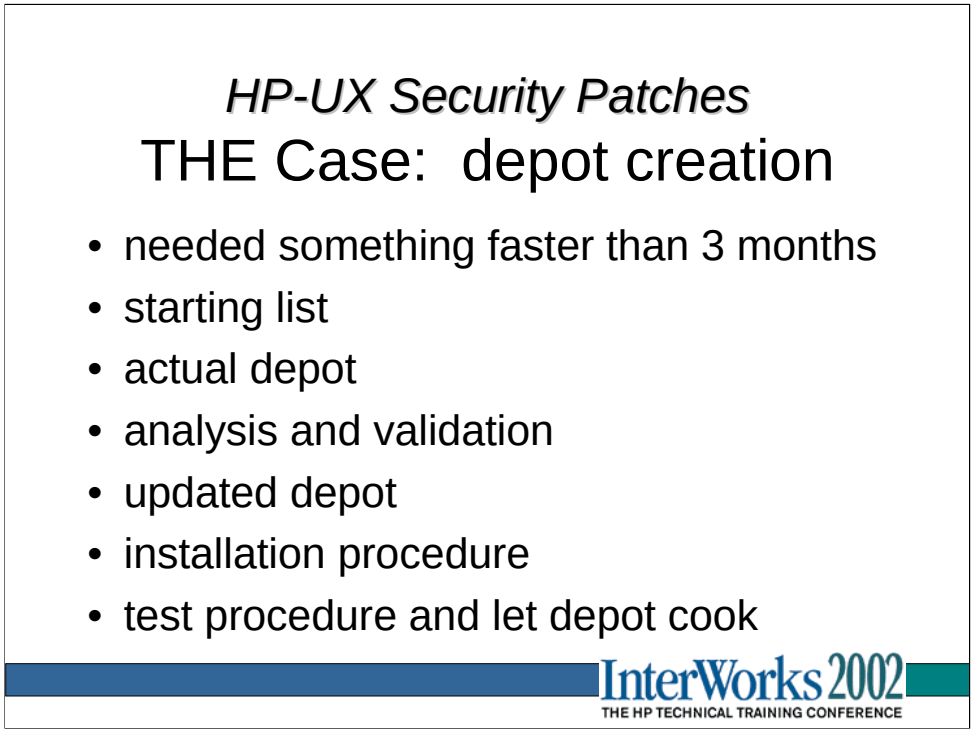
“unreasonable” deadline extended

•deadline was extended to mid-March 2002

HP-UX Security Patches

THE Case: external resources

- HP Response Center -- to create security patch tape (once and done, not regular service)
- CSS consultant engaged -- detailed patch analysis/validation; answer our questions
- sys admin contractors -- to augment the patch deployment team
- non-UNIX personnel -- for project coordination, administration, and paperwork



needed something faster than 3 months

•March 2001 depot took 3 months to create

•this fast-track project required something quicker

•was addressed by additional resources, focused and full-time efforts, expenditures for additional time & materials, and sacrifice of some other projects

starting list

•HP ITRC home page --> maintenance and support

• --> support information digests

• --> security bulletin archive

• --> HP-UX patch security matrix

• --> current patches for security issues

actual depot -- obtained on tape from the HP Response Center

analysis and validation

•read patch text

•HP ITRC customized patch analysis

•get any questions answered

•stay within our desired conservative approach

•create work-arounds for any issues uncovered

updated depot

•containing all corrections, additions, supercedes, and dependencies

•set depot up on network-connected server

create procedure for installing patch depot

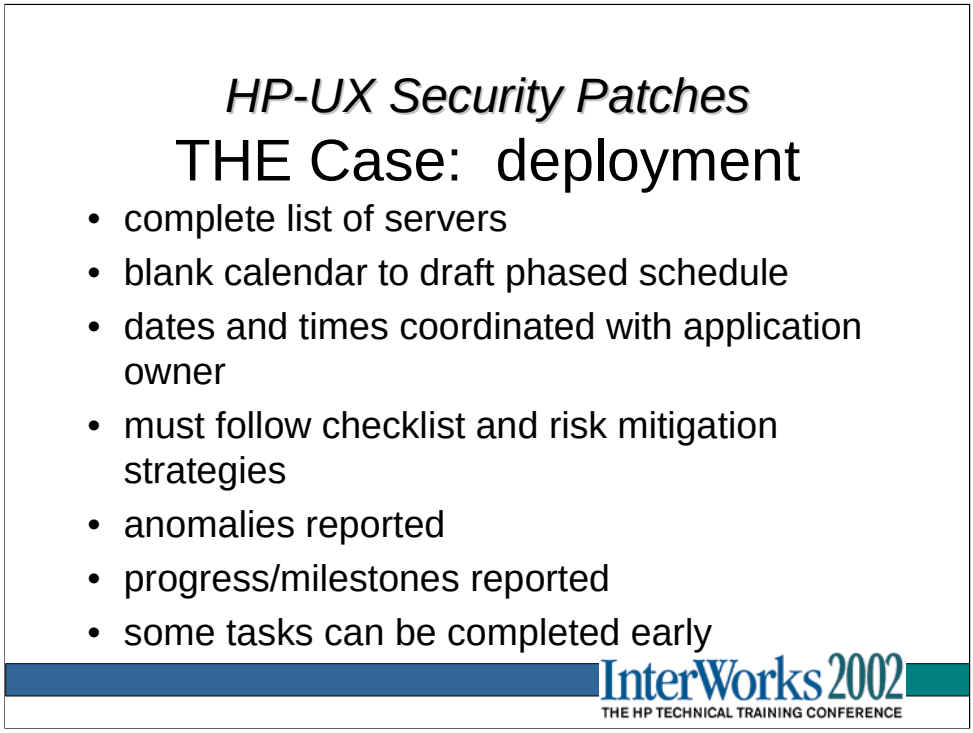
•assure server is at March 2001

•pre-requisite patches

•the rest of the security patches

•any required work-arounds

test the procedure and let depot cook -- preferably for 1 week and under load



1. need complete list of servers and their current OS version

2. starting with blank calendar, draft phased schedule of patch deploymenht

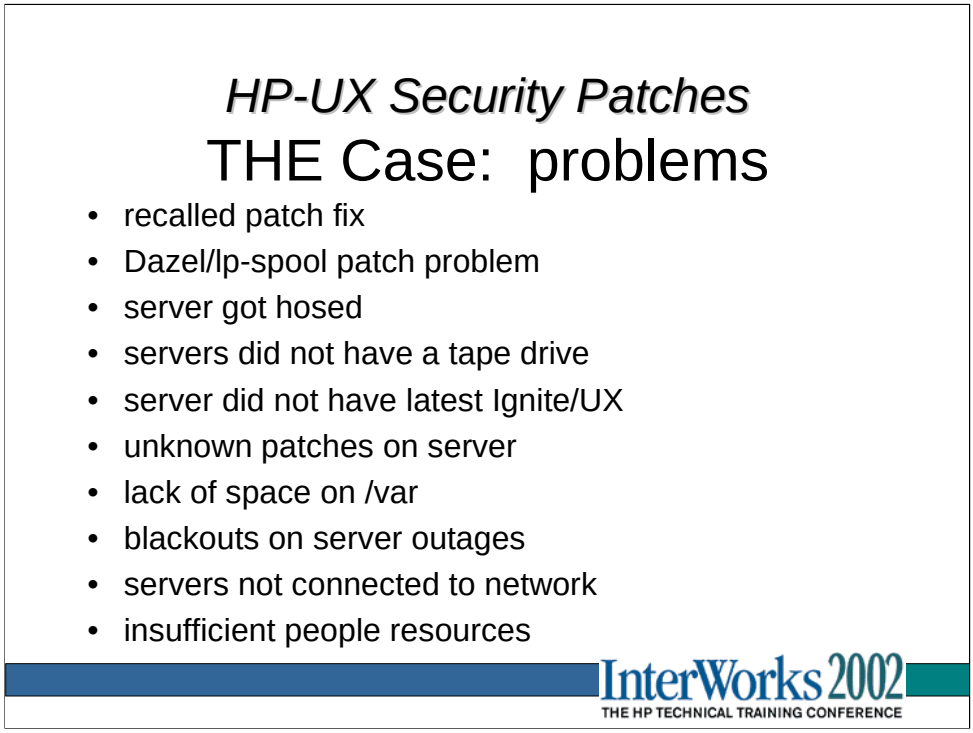
3. actual dates and times coordinated and confirmed with the server(s)' application owner

4. must follow the checklist and all approved risk mitigation strategies

5. anomalies are reported back to the depot creation team

6. progress/milestones reported are reported back to the sponsor team

7. some tasks can be completed early -- pre-analysis, swinstall/analysis, disk space analysis, make-recovery backup, pre-patch reboot can all be completed days to weeks prior to the actual patch depot install



recalled patch fix

•along the way, March 2001 had some recalled patches that could not be superceded

•old patch had to be removed prior to applying new patch

•required manual steps to accomplish

Dazel/LP-SPOOL patch problem

•Dazel product replaces native "lp" command

•however, LP-SPOOL patch replaces "lp" command

•first 11 servers that were rushed broke printing

•required fixup procedure for those 11 servers

•required extra steps in "flight checklist" to disable DAZEL prior to installing patch depot

server got hosed -- fortunately, make-recovery backup successfully restored it

servers did not have a tape drive

•we had a roving tape drive

•for those servers where attaching tape drive was impracticable, we made sure they had latest Ignite/UX, and used make-net-recovery

server did not have latest Ignite/UX -- always updated server to latest Ignite/UX

unknown patches on server -- created "get-swinfo" script (see RISK MITIGATION)

lack of space on /var -- created space on another drive and symlinked from /var/adm/sw to new space

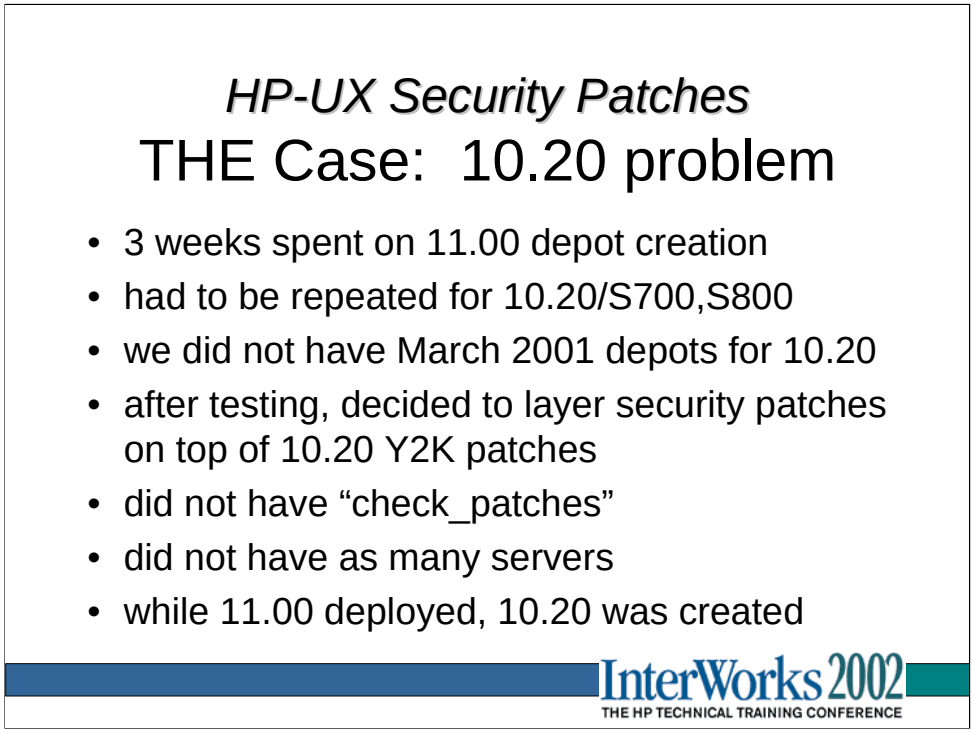
blackouts on server outages -- documented and deferred until after outage blackout

servers not connected to network -- created tape depot

insufficient people resources

•one group of consultants stated that it would take a full-time dream team of 6 people 3 weeks to create one patch depot

•another company said they had only 2 people for depot creation, and 200 people on their troubleshooting swat team



1. 3 weeks were spent on the HP-UX 11.00 depot creation

2. we had to be repeat this twice for 10.20 -- one for S700 and one for S800

3. we did not have March 2001 depots for 10.20, only the installed Y2K depots

4. after testing it, we decided simply to layer the security patch depot on top of the 10.20 Y2K patches

5. we did not have the same tools available such as “check_patches”

6. fortunately, did not have as many HP-UX 10.20 servers

7. while 11.00 patches were being deployed, the 10.20 depots were being created

1. Nearly 200 servers were patched to the November 2001 security patch level.

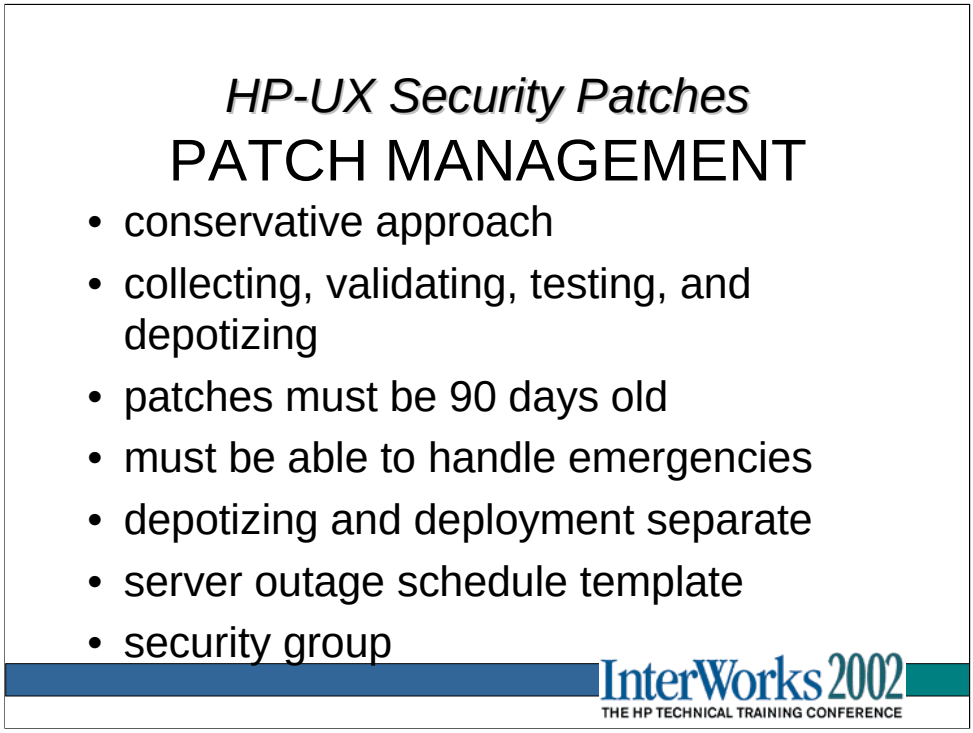
2. Created depot about every 3 weeks. Previously took almost 3 months (for the March 2001 depot).

3. Deployment took about 2 months. Previously took about 5 months for the Y2K patch deployment.

4. This was an extraordinary effort, and is not repeatable with our present resources and infrastructure.

5. Therefore, a project team was formed to recommend an infrastructure and procedure for periodic and emergency patch management.

6. Now, 6 months later, that project is just about finished.



1. conservative approach; not innovative or restrictive

2. collecting, validating, testing, and depotizing of patches

3. patches must be 90 days old

4. must be able to incorporate emergency patches (that includes security patches); patches may be younger than 90 days

5. creation of patch depots and their deployment must be separate activities with possibly separate teams

6. server outage schedule template is that which we used for the security patching fire drill

7. security group is responsible for dispositioning alerts and tracking security patches.

8. more details available once we finish the project

HP-UX Security Patches

FINAL THOUGHTS

- Methodical approach
- Risk mitigation strategies
- Flight checklist
- Abort if any problems are encountered; don't try to wing it.
- Have fun! (Why does every presenter say that?)