Progresswiz

From Vulnerable to Inviolable

How Company X Secured Their OpenEdge Environment

PAUL KOUFALIS
PRESIDENT
PROGRESSWIZ CONSULTING

In 50 Slides or Less…

- You will learn:
  - What happens in the real world when a company takes security seriously
  - The “Before” view
    - What 90% of OE environments look like today
    - Yours maybe?
  - The “After” view
  - And all the steps in between

Progresswiz

Paul Koufalis? Who is Paul Koufalis?

- Famous quote from my junior college days
  - Buy me a beer and I'll tell you the story
- Progress DBA and UNIX sysadmin for 18 years
- Security has become a bigger and bigger part of my practice over the past few years

Progresswiz

Let’s Get Started

- First a quick intro
- My multi-phase approach to securing the customer’s environment
- What’s next for customer X
- Questions

Progresswiz

Introduction

- This is a real life implementation
- Customer thought they were *much* more secure than they actually were
- I see this over and over again…

- Don’t let me look at your environment unless you want to hear the truth

Progresswiz

What is “Secure” ?

- You tell me
  - Not the same answer for everyone
- Checkpoint Firewall or D-Link Router “Firewall” ?
- Passwords or RSA SecurID token?
  - or iris scans?

Progresswiz

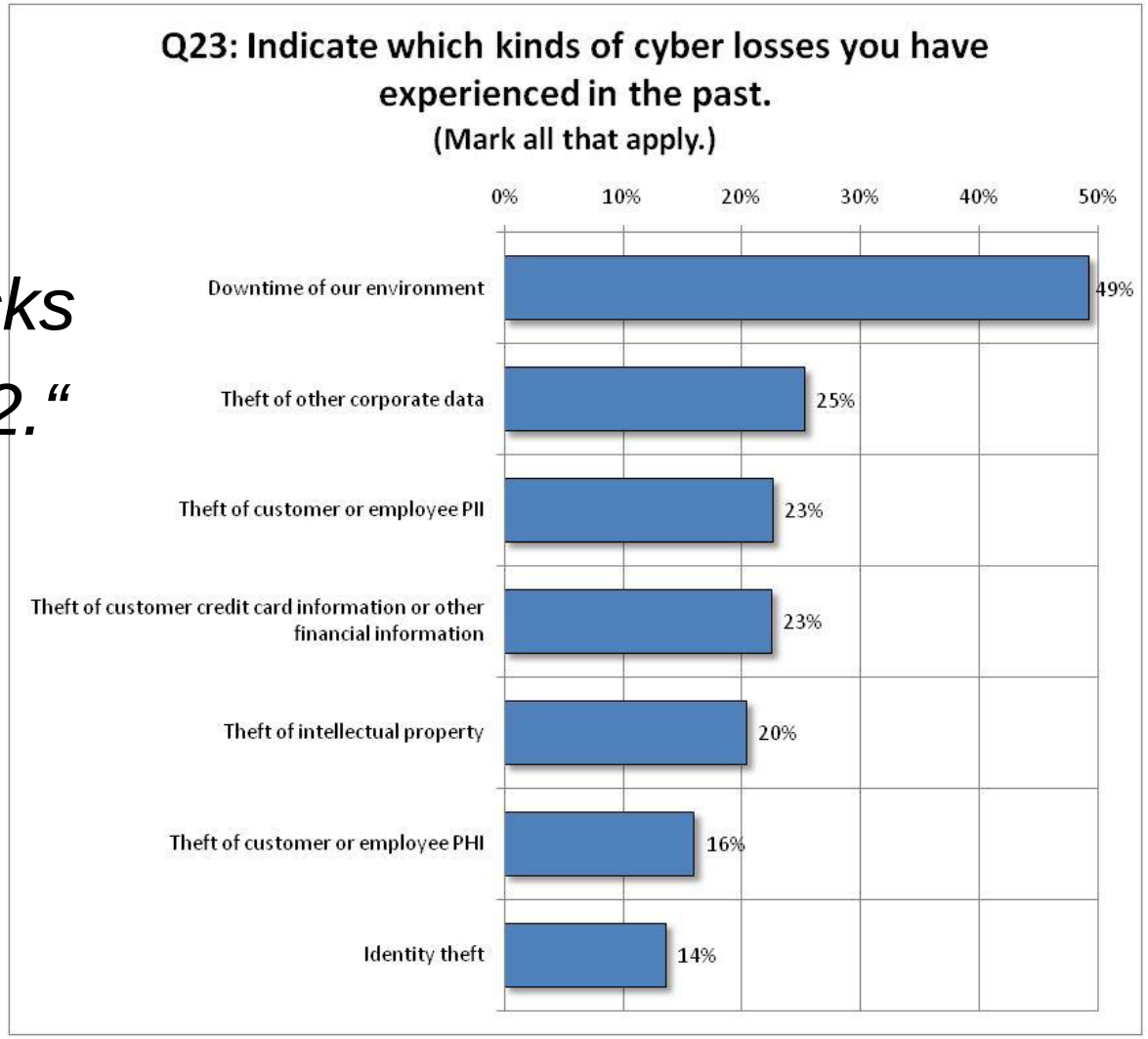
Complete Security

- **Does not exist**
- RSA hacked in March 2011
  - SecurID token info stolen
- Sony hacked in April 2011
  - Personal info from 100M people stolen
- Citigroup hacked in June 2011
  - Customer info stolen
- YOU in May 2012
  - Would you even know if someone was stealing your data?

Progresswiz

Symantec SMB Survey

"…the average annual cost of these cyber attacks was US $188,242."



Progresswiz

Think Layers

- Good security is layer after layer of annoyance for the bad guy
- Aim for many layers of security
- Think “Peeling an onion”

- See “Securing Your Swiss Cheese Environment” for more on this

Progresswiz

ROI

- Look for the most bang for your $$
  - That includes professional services
- So much of this stuff is ridiculously easy and cheap to implement
- Pick the low hanging fruit first
- Now let’s get started…

Progresswiz

“Old” PROD

- DBs on three UNIX servers
- GUI, ChUI and batch ABL clients
- SQL clients
  - Internal and external (web)
- Nothing encrypted
  - telnet, ftp, rcp/rsh

Progresswiz

“Old” PROD

- All source code everywhere
- Almost no security in the DB
  - Some SQL grants
  - “!,*” in _CAN-*
- _User records existed
- Batch passwords widely known
  - Readable PF files with “–U batch –P batch”
- Full dev license installed in PROD
- File permissions mostly wide open

Progresswiz

“Old” PROD

- Customer thought their data was secure
  - They had users and passwords, right !?!
- I extracted data
  - With no user/password
  - From a session on the local server
  - From a session on a remote server
  - I could have used “strings” on DB files
- Oops…

Progresswiz

Phase I - Locking Down New Environment

- Low hanging fruit first
- Cheap and easy to implement

- You have no excuse to not implement these changes at your site

Progresswiz

Database Server

- No more root
  - No one has the password
  - Locked in a drawer
- Sudo for everything
  - BUT no one allowed to “sudo su - <user>

Progresswiz

Database Server

- Change owner and perms of all users' .profile
  - Easy to ftp in, get .profile, modify, put back
  - Ta-da! Shell access
  - I SEE THIS ALL THE TIME!!! ARRGGHH!!!
- Use "exec" to start application for regular users
  - Don't use "trap" as it could cause application problems

Progresswiz

Database Server

- Create user prodba + group dba
  - No one has the password
- All DB files belong to prodba/dba
- Only user prodba can start/stop/etc DBs

- With prodba:dba you can strictly control who has physical access to the DB files
  - Without TDE, physical access = data access
  - Hacking out table security is easy

Progresswiz

Database Server

- DBA's login as themselves and use sudo
  - They are NOT members of dba group

```
pkoufal@srv> sudo –u prodba
                    /usr/local/bin/startdb.ksh
```

- Sudo actions logged:

```
Sep 11 19:57:45 bart sudo: pkoufal : TTY=pts/0 ;
   PWD=/home/pkoufal ; USER=prodba ;
   COMMAND=/usr/local/bin/startdb.ksh
```

  - You always know who did what and when

Progresswiz

Database Server

- Database directory 750:

```
-rwxr-x--- prodba dba    /db
```

- All Database Files 660:

```
-rw-rw---- prodba dba    /db/toto.db
```

- Group members can modify files but not create/delete
- These perms ok for regular users
  - On UNIX _progres has setuid bit

Progresswiz

Example

- Given:

```
-rwxr-x--- prodba dba   /db
-rw-rw---- prodba dba   /db/toto.db
```

```
pkoufal@srv> ls /db/toto.db
toto.db not found

pkoufal@srv> _progres /db/toto
There is no server for database /db/toto. (1423)
```

  - Works fine

Progresswiz

$DLC

- Segregate the full dev license

```
-rw------- prodba dba full.cfg
-rw-r--r-- prodba dba progress.cfg
-rw-r----- prodba dba qr.cfg
-rw-r--r-- prodba dba runtime.cfg
```

- Progress.cfg = runtime.cfg
- Query avail for support (via sudo)
- Runtime available to all

Progresswiz

Query-Only

- Big issue: support needs ad hoc access
  - Read only of course
- Wrote script proqr.ksh
  - export SHELL=/usr/local/bin/proqr.ksh
  - export PROCFG=$DLC/qr.cfg
  - _progres /db/toto …
- Use sudo to run:

```
pkoufal@srv> sudo –u proqr qr.ksh
```

Progresswiz

Query-Only

- export SHELL=… not perfect
  - User cannot OS-COMMAND nor Tools – OS Shell
  - Can still OS-COPY, OS-DELETE…
- Force keyword forget (-k keyforget_file)
  - “Forget” all the OS-* ABL keywords
  - User would need access to malicious r-code to do damage
- Another possibility would have been DBAUTHKEY to lock down r-code
  - Perhaps as a future enhancement

Progresswiz

Application

- Regular users in group named after app
- Code and directories read-only to app group
  - RW to owner

```
drwxr-x--- app appgrp /myapp
-rw-r----- app appgrp /myapp/menu.p
```

  - Note ".p" and not ".r" – more later

Progresswiz

End – Phase I

- Good start – easy and cheap as promised
- Low disruption risk

- Result: If you are on the server your actions are fairly limited

Progresswiz

Phase II

- Required a little more work and planning
- Still fairly easy and inexpensive
- Could be disruptive if not planned and executed properly

Progresswiz

Database Server

- Shut off non-secure services
  - ftp, telnet, rsh, rlogin, etc…
- This had the potential to cause some headaches
  - Multiple scripts existed that used ftp
  - Had to replace old telnet-only emulators

Progresswiz

Database Server

- Changed all scripts that use ftp to use scp
  - Setup “authorized_keys” to eliminate passwords in scripts
- Changed all rsh jobs to ssh
- Changed all terminal emulators to ssh
  - Putty for support people
  - Commercial product for users

Progresswiz

Database Server

- Start all brokers with –S 4GL or –S SQL
- Start all brokers with –minport and –maxport
  - Keep the range tight
  - Keep the range outside O.S. port range for outbound connections if large # of servers
- Will be needed later for network changes

Progresswiz

$DLC

- Lock down $DLC files on all servers
- Somewhat complex
- Example changes:
  - chown –R prodba:dba $DLC
    - Careful: setuid execs must stay root
  - chmod –R o-r $DLC/src and $DLC/tty
    - Again careful with some files
- THIS IS NOT A COMPLETE LIST OF CHANGES. DO NOT PLAY IN $DLC!!

Progresswiz

$DLC

- Remember that $DLC contains
  - Encryption certificates and keys
  - Ubroker.properties etc. files
- At the very least make these read-only for regular users

Progresswiz

$DLC

- Unauthorized users no longer can:
  - Backup the db
  - Binary dump data
  - Run the Editor or Data Admin Tool
  - Almost everything prohibited unless member of dba group

Progresswiz

Database

- Sounds dumb but REAL usernames and passwords
  - Half the I.T. department knew the username/password for batch jobs
  - Generic “odbc” username/password used by ALL SQL connections
    - GRANTed SELECT on all tables
    - And of course everyone knew that password too

Progresswiz

Database

- Real passwords saved in secure PF file
  - In OE11 can use encrypted password
- Batch jobs can be started with sudo
- Support cannot read the PF file

- Create real SQL users
  - Each has only the required GRANTs
  - No more lazy GRANT SELECT to PUBLIC

Progresswiz

Database

- Create _USER record for sysprogress
  - Often overlooked
  - Important note: This user is not meant to be used by users for querying etc.
    - For admins only!
- Check SYSDBAUTH
  - Sysprogress assigned DBA and resource roles
  - Also user that created database

Progresswiz

ODBC Passwords

- On UNIX lock down odbc.ini
  - Contained batch passwords
- On Windows the user uses his application username/pwd
  - NOT a generic username/pwd

Progresswiz

End – Phase II

- Was not super-easy but not that hard
- Cost was extremely low

- Result: Server environment very secure

Progresswiz

Phase III - Future

- Most are not necessarily difficult
- But they are “scary” to management
  - Can definitely disrupt business if done incorrectly
- If you want to use these, plan carefully and test extensively

Progresswiz

VLAN Segregation

- PROD servers should be in their own VLAN
- General rule:
  - If you have access to a full.cfg you do not have access to PROD DB ports
  - I.e. PCs with OE Studio etc…
  - Dev UNIX servers

Progresswiz

VLAN Segregation

- I am not a network guy but this is supposedly easy:
- Access list on target interface into PROD VLAN
  - Allow only PROD user traffic to DB ports
  - Drop all other requests to DB ports
  - Does not block other traffic (ex.: ssh)
- Newer NICs have built-in VLAN support

Progresswiz

Application Code

- Still p-code in PROD
  - Officially a licence violation since running with full dev license
  - And not very secure!
- Resistance to r-code only in PROD
  - New = different => apprehension
  - Understandable
- Remove “.” from PROPATH
  - Disaster waiting to happen

- Use DBAUTHKEY to further secure r-code

Progresswiz

Application Code

- Deployment difficulties are more logistical than technical
  - Need to get SOPs in place
  - Implement source code control and r-code deployment structures
  - Already used for other applications so just need to apply to OE environment

Progresswiz

DB Security

- Enable runtime security
- Technically easy but tricky
  - Which user gets what access?
  - If you screw it up you will have some very angry users!
- At the very least disable blank userid access + runtime security
  - Valid users can do everything
  - Force them to run r-code to limit access

Progresswiz

DB Security

- Create Security Administrators
  - DB Administration Tool
  - Only SecAdmins can play in schema tables
- Careful: New tables/fields do not inherit SecAdmin info in _CAN* fields

Progresswiz

Data

- Scramble data before refreshing DEV/TEST
  - Tools exist – it's just a question of picking one and implementing it
- This one often slips through the cracks
  - People trust their I.T. department

Progresswiz

End – Phase III

- Once done it will be very difficult to access OpenEdge data without a valid username and password
  - From anywhere
- The weakest link is still the username and password
  - Not surprising

Progresswiz

Phase X

- Unplanned Future Stuff
- Authentication with CLIENT-PRINCIPAL
  - LDAP – Active Directory
  - Current logged-in account
- In the DB:
  - Disable Blank UserID Access
    - DB Options – NOT DB Security

Progresswiz

Final Thoughts

- Remember: security is layers:
  - Secure access to systems
    - Built-in features in OpenEdge and O.S.
  - Encrypt data and communications
    - SSL
    - OpenEdge Transparent Data Encryption
  - Audit users
    - OpenEdge Auditing

Progresswiz

Progresswiz Consulting

- Based in Montréal, Québec, Canada
- Providing technical consulting in Progress®, UNIX, Windows, MFG/PRO and more
- Specialized in
  - Security of Progress-based systems
  - Performance tuning
  - System availability
  - Business continuity planning

Progresswiz

Questions?

Progresswiz

Progresswiz Consulting

- Further questions or comments? Send me an email:

  pk@progresswiz.

Progresswiz