Network Nightmare

Intel PXE

This Talk

- Whoami
- Motivation
- PXE
- Attacks
- Defense
- Lessons Learned

#whoami

- Matt Weeks
- MSF, math, exploits, crypto
- Scriptjunkie if you hang out on irc
- @scriptjunkie1
- http://www.scriptjunkie.us/
- scriptjunkie {shift+2} scriptjunkie.us

What’s going on here

- Hacker’s perspective
- Attacks
  - Design
  - Exploit

Exploits

- Unpatched vs. Patched
- Remote vs. Local
- Popular vs. Obscure
- Root vs. User
- Unauthenticated vs. Authenticated
- Reliable vs. Unreliable
  - Version specificity
  - OS protections

Exploits

- Published
- 0-day
  - Time
  - Fuzzing/static analysis -> Vulnerability ID -> exploit -> Bypass protections -> … escalate privs

Easier way?

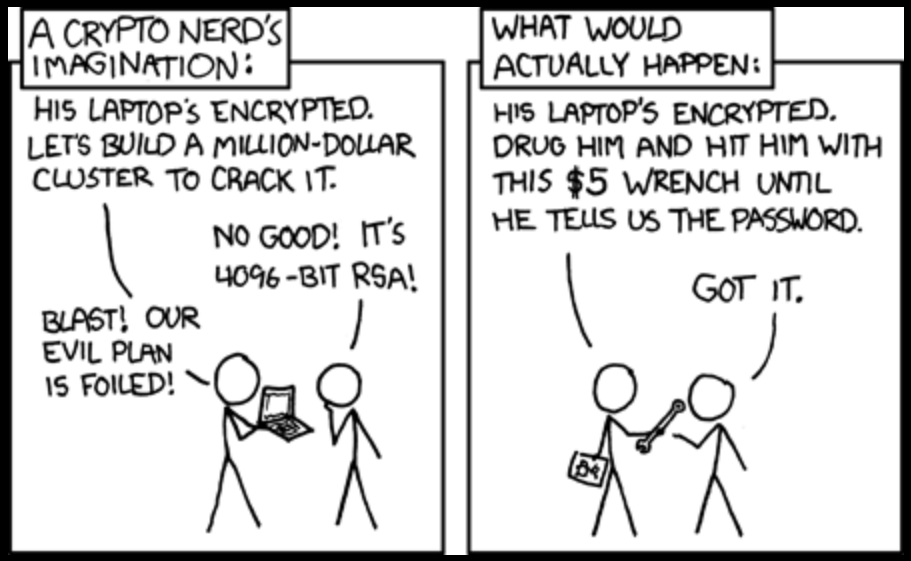
- Offline attack

Offline attacks

- Evil maid attack

Offline attacks

- Evil maid attack
- Rubber hose cryptanalysis



XKCD - http://xkcd.com/538/

Downsides

- Physical access
- Unstealthy
- Jail?
- Still common


PXE
[preboot execution environment]

PXE

- “network boot”
- Protocol+firmware to boot from NIC
  - Introduced by Intel and SystemSoft
- DHCP & TFTP based
- BIOS-level access
  - Full system control
  - Bypass host hardening/OS/AV
  - OS-agnostic
  - Network!

How it works

- Step 1 – Your computer shuts down

How it works

- Step 2 – Wake up … something's different

(*The Godfather*) © 1972 Paramount Pictures

PXE Proliferation

- Every BIOS (almost)
- How widely enabled?
- I have seen PXE …
  - left on
  - occasionally enabled & used
  - turned off

Just to be clear

PXE

PXE

- **Unpatched** vs. Patched

PXE

- **Unpatched** vs. Patched
- **Remote** vs. Local

PXE

- **Unpatched** vs. Patched
- **Remote** vs. Local
- **Popular** vs. Obscure

PXE

- **Unpatched** vs. Patched
- **Remote** vs. Local
- **Popular** vs. Obscure
- **Root** vs. User

PXE

- **Unpatched** vs. Patched
- **Remote** vs. Local
- **Popular** vs. Obscure
- **Root** vs. User
- **Unauthenticated** vs. Authenticated

PXE

- **Unpatched** vs. Patched
- **Remote** vs. Local
- **Popular** vs. Obscure
- **Root** vs. User
- **Unauthenticated** vs. Authenticated
- **Reliable** vs. Unreliable
  - **Version independent** not version specific
  - **Bypasses** OS protections

Why oh why?

- Top syadmin reasons:

Why oh why?

- Top syadmin reasons:
  - Image deployment

Why oh why?

- Top syadmin reasons:
  - Image deployment
  - System restoration

Why oh why?

- Top syadmin reasons:
  - Image deployment
  - System restoration
  - Just in case

Why oh why?

- Top syadmin reasons:
  - Image deployment
  - System restoration
  - Just in case
  - What's that? I have that on?

How PXE works

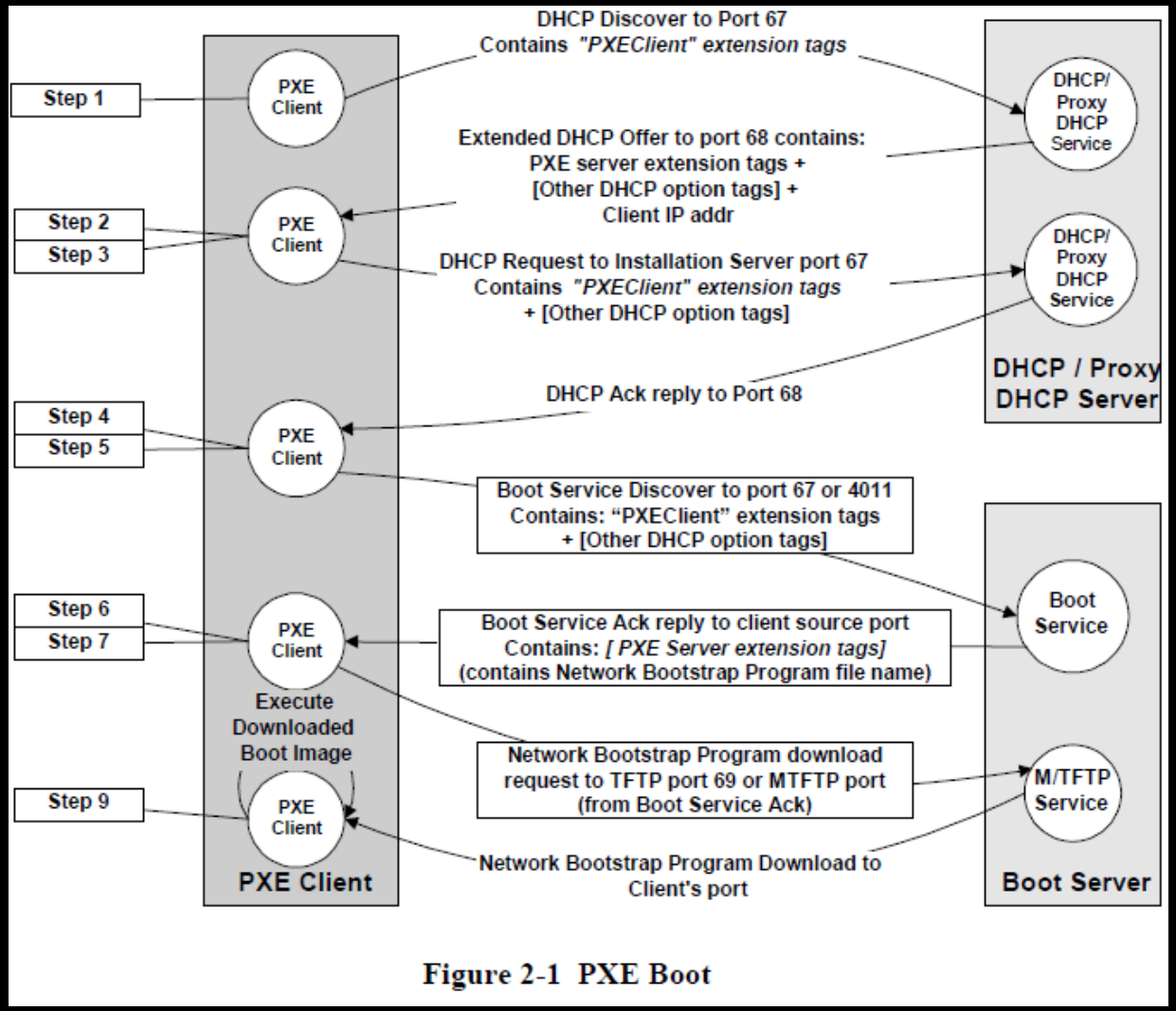


Figure 2-1 PXE Boot

Preboot Execution Environment (PXE) Specification 2.1 Intel Corporation/SystemSoft

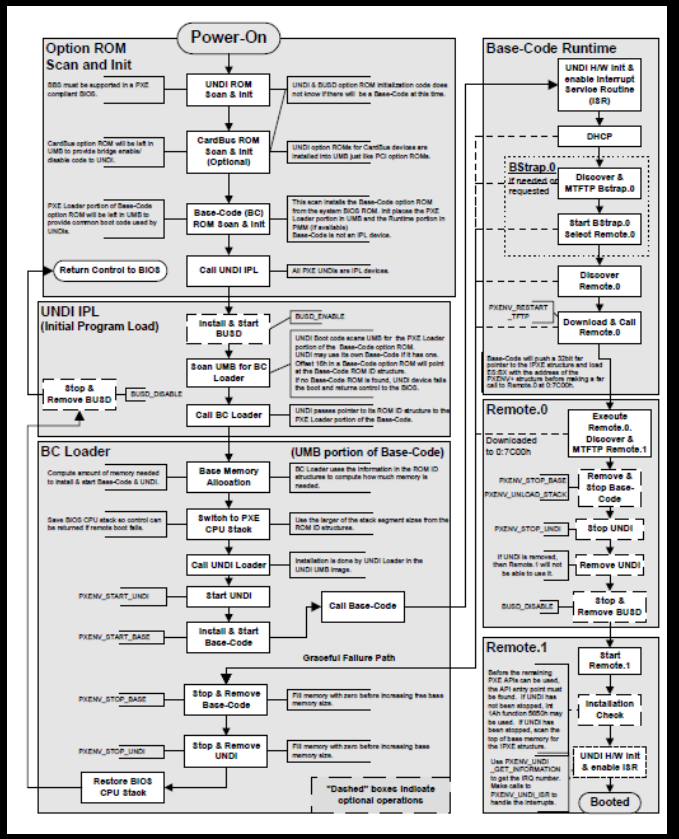
PXE Difficulties

- Passive
  - Wait
  - Wake-on-LAN
- DHCP
  - Race condition
- TFTP
- Code execution
  - Bare metal

PXE Difficulties

Preboot Execution Environment (PXE) Specification

Version 2.1



Previous PXE “attacks”

- Not attacks
- Admin tools
  - Imaging
  - Pxelinux

Previous PXE “attacks” - Imaging

- Requires server software
- Time-consuming
- Imaging =

Previous PXE “attacks” - PXELinux

- Manual PXE server
- Manual DHCP
- Difficult to deploy remotely
- Unreliable or lack targets
- Lack custom payloads


Online Control

Online Control

- PXE-bootable Linux live CDs
  - DSL
  - Tiny Core
  - Knoppix
- Strategy
  - Remaster live CD
  - Boot via PXE
  - Scripts connect back
  - Shell!

Online Control Demo

Online Control Advantages

- Any OS
- Flexibility
- No prior knowledge

Online Control Problems

- MyNetworkCard™ drivers
  - Distro != initrd
- Time
  - Human in the loop
  - Visual indicators

Offline Code Injection

Offline Code Injection

- Inevitable
- Admin privileges
  - Inside > outside

Offline Linux Code Injection

- Shellcode on boot
  - Write/edit file to RCE
    - /etc/init.d/…
    - ~/.bashrc etc
- User add
  - /etc/passwd
  - ~/.ssh/authorized_keys

Offline Windows Code Injection

- Bootkits
- Binary planting
- Binary swapping
- Binary embedding/modification
- DLL preloading
- Registry edits
- Binary swapping + service editing

Note!

- This presentation will not be addressing FDE
- See cold boot or evil maid

Bootkits

- Sinowal
- Stoned
- Whistler
- TDL/Alureon
- eEye BootRoot (PiXiE)

Bootkits: Advantages

- Awesomeness
- Stealth
- Privileges

voxx - Belchfire

Bootkits: Disadvantages

- OS-specific
- OS protections
- Work factor

Binary Planting

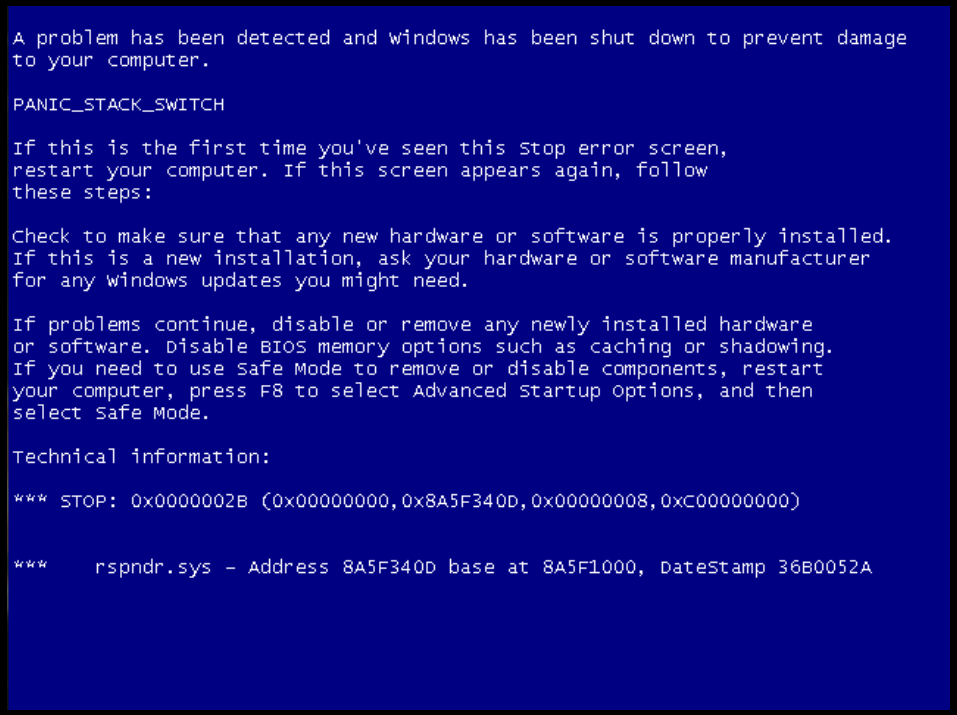
- Startup folders
  - C:\Documents and Settings\All Users\Start Menu\Programs\Startup
  - C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup
  - Unprivileged
- WBEM .mof method
  - **Stuxnet!**
  - No Vista+

Binary Swapping

- Example:
  - Swap services/svchost/wininit/…
  - Spawn old exe and payload
  - Swap back
- Advantages:
  - Guaranteed RCE
  - Portable

Binary Swapping Problems

- Bluescreens
  - Early processes bluescreen on exit
  - Cleanup requires exit
- Disabled services
  - Late process non-critical
  - Spoolsv.exe …



Binary Embedding/Modification

- Inject additional code into existing .exe files
  - svchost/wininit/winlogon/…
- Example:

  msfvenom -f exe -x svchost.exe -k -p - < pay > a.exe

Binary Embedding Problems

- Architectures
  - x86 != x64
- Slack space
- Cleanup

DLL Preloading

- Swap system dll
- Add dll higher in search path
- Problems:
  - Architecture
  - Imports
- Still an option

Registry Edits

- Lots of options!
  - Run keys - HK(LM|CU)\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    - Reliable
    - Unprivileged
  - Service addition HKLM\SYSTEM\CurrentControlSet\Services
    - Privileged!
    - OS version differences

Registry Edits

- Service Editing HKLM\SYSTEM\CurrentControlSet\Services
  - Privileged!
  - Change binpath string
  - Check type, start
- Known DLL's
  - Privileged!
  - Add string
- And others

Registry Edits

- Linux initrd
- Adding registry data
- Chntpw's ntreged library
- Warning …
- HKLM corruption =

  game over

Photo from Cleveland.com

Binary Swapping + Regedit

- Swap a non-essential service
- Edit DWORD start value
- Profit

Binary Swapping + Regedit

- Reliable
- No bluescreens
- Cross-arch
- No registry corruption warnings


Pivoting

Pivoting

- LAN-hop
- Meterpreter-in-memory
  - Railgun
    - Network delay
  - Extension
    - Compiled program

Meterpreter Review

- TLV request
- Embedded DLL
- Reflective Loader
- Method Calls

Attack Recap

1. Dynamic payload generation
    1. [wake-on-LAN]
2. DHCP
3. TFTP
4. PXELinux kernel, initrd
5. Swap
6. Registry
7. Reboot
8. Payload
9. Cleanup

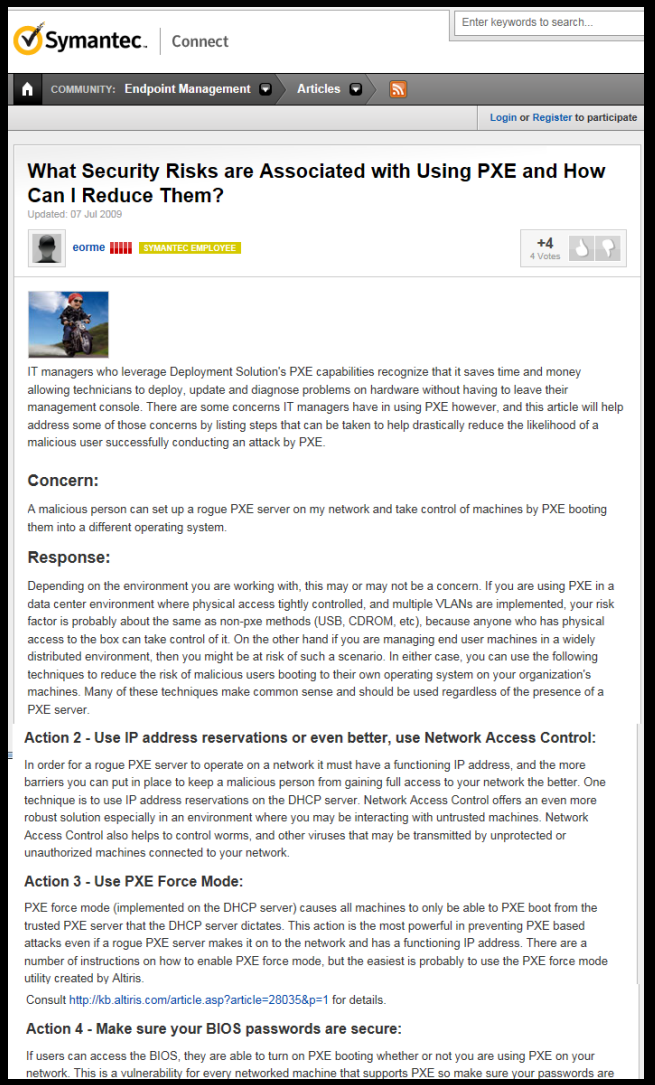
Metasploit Demo

Compatibility


Defense

Defense: Fail

- Bad advice
- How to fail:
  - IP reservations
  - NAC
  - PXE Force Mode
  - BIOS passwords



Symantec. | Connect

Enter keywords to search...

COMMUNITY: Endpoint Management | Articles

Login or Register to participate

What Security Risks are Associated with Using PXE and How Can I Reduce Them?

Updated: 07 Jul 2009

eorme SYMANTEC EMPLOYEE

+4 4 Votes

IT managers who leverage Deployment Solution's PXE capabilities recognize that it saves time and money allowing technicians to deploy, update and diagnose problems on hardware without having to leave their management console. There are some concerns IT managers have in using PXE however, and this article will help address some of those concerns by listing steps that can be taken to help drastically reduce the likelihood of a malicious user successfully conducting an attack by PXE.

Concern:

A malicious person can set up a rogue PXE server on my network and take control of machines by PXE booting them into a different operating system.

Response:

Depending on the environment you are working with, this may or may not be a concern. If you are using PXE in a data center environment where physical access tightly controlled, and multiple VLANs are implemented, your risk factor is probably about the same as non-pxe methods (USB, CDROM, etc), because anyone who has physical access to the box can take control of it. On the other hand if you are managing end user machines in a widely distributed environment, then you might be at risk of such a scenario. In either case, you can use the following techniques to reduce the risk of malicious users booting to their own operating system on your organization's machines. Many of these techniques make common sense and should be used regardless of the presence of a PXE server.

Action 2 - Use IP address reservations or even better, use Network Access Control:

In order for a rogue PXE server to operate on a network it must have a functioning IP address, and the more barriers you can put in place to keep a malicious person from gaining full access to your network the better. One technique is to use IP address reservations on the DHCP server. Network Access Control offers an even more robust solution especially in an environment where you may be interacting with untrusted machines. Network Access Control also helps to control worms, and other viruses that may be transmitted by unprotected or unauthorized machines connected to your network.

Action 3 - Use PXE Force Mode:

PXE force mode (implemented on the DHCP server) causes all machines to only be able to PXE boot from the trusted PXE server that the DHCP server dictates. This action is the most powerful in preventing PXE based attacks even if a rogue PXE server makes it on to the network and has a functioning IP address. There are a number of instructions on how to enable PXE force mode, but the easiest is probably to use the PXE force mode utility created by Altiris.

Consult http://kb.altiris.com/article.asp?article=28035&p=1 for details.

Action 4 - Make sure your BIOS passwords are secure:

If users can access the BIOS, they are able to turn on PXE booting whether or not you are using PXE on your network. This is a vulnerability for every networked machine that supports PXE so make sure your passwords are

http://aka-community.symantec.com/connect/fr/articles/what-security-risks-are-associated-using-pxe-and-how-can-i-reduce-them

Defense: Less Fail

- DHCP Attack Detection
  - Scan
  - Check for duplicate replies
  - Check for ARP poisoning
  - Check for unregistered clients

Defense: Good Idea

- Firewalls
  - Only allow DHCP traffic to/from server
  - ARP poisoning

Defense: Better Idea

- VLAN isolation
  - Multiply VLANs
  - Limit broadcast domains
  - Forward DHCP traffic
  - Use switch/routers

Defense: Great Idea

- Boot Integrity Services
  - + PXE extension
  - + Signed code
  - – Certificate deployment
  - – Compatible firmware

Defense: Best Idea

- Turn it off


Lessons Learned

Protocol Design

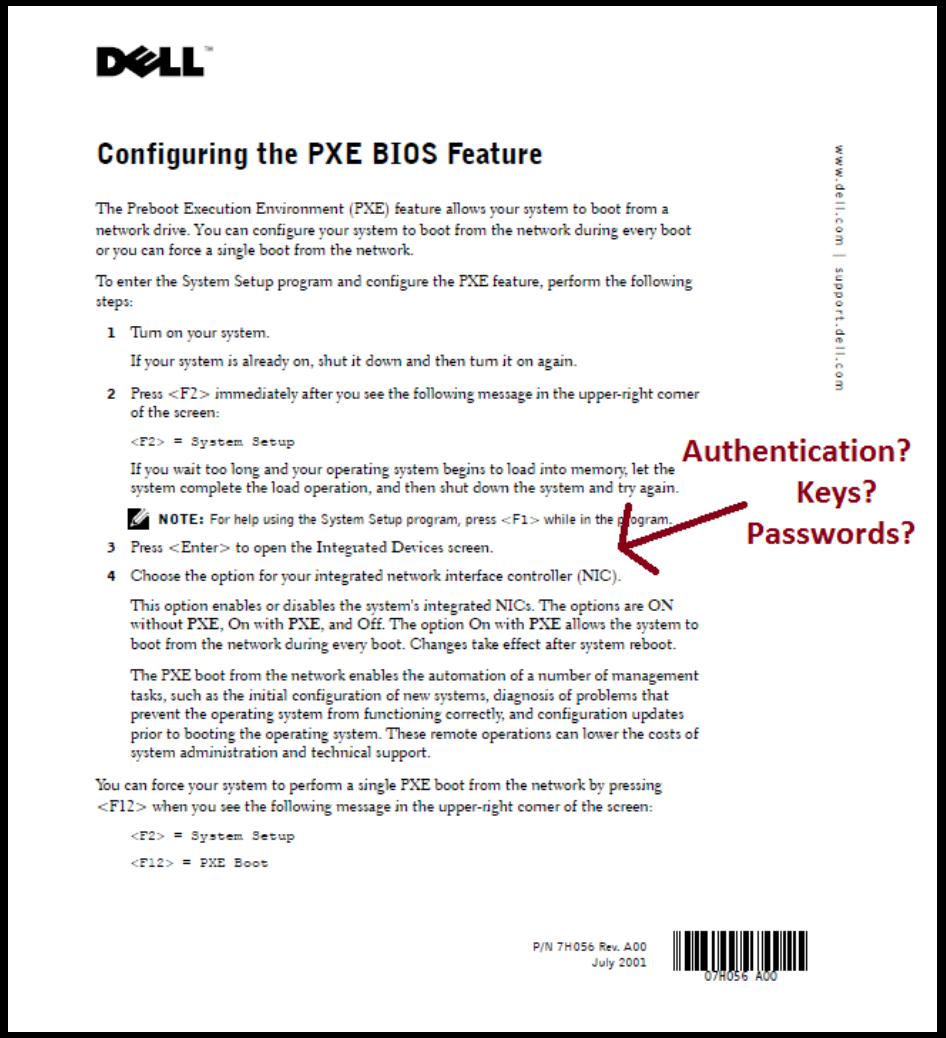
- Security from start
- Authentication
- Access Controls
- Attack it

Implementation

- Security by default
- Protest insecurity

Documentation

- Instruct your users
- Secure how-to guides



DELL

Configuring the PXE BIOS Feature

www.dell.com | support.dell.com

The Preboot Execution Environment (PXE) feature allows your system to boot from a network drive. You can configure your system to boot from the network during every boot or you can force a single boot from the network.

To enter the System Setup program and configure the PXE feature, perform the following steps:

1 Turn on your system.

If your system is already on, shut it down and then turn it on again.

2 Press <F2> immediately after you see the following message in the upper-right corner of the screen:

<F2> = System Setup

If you wait too long and your operating system begins to load into memory, let the system complete the load operation, and then shut down the system and try again.

NOTE: For help using the System Setup program, press <F1> while in the program.

3 Press <Enter> to open the Integrated Devices screen.

4 Choose the option for your integrated network interface controller (NIC).

This option enables or disables the system's integrated NICs. The options are ON without PXE, On with PXE, and Off. The option On with PXE allows the system to boot from the network during every boot. Changes take effect after system reboot.

The PXE boot from the network enables the automation of a number of management tasks, such as the initial configuration of new systems, diagnosis of problems that prevent the operating system from functioning correctly, and configuration updates prior to booting the operating system. These remote operations can lower the costs of system administration and technical support.

You can force your system to perform a single PXE boot from the network by pressing <F12> when you see the following message in the upper-right corner of the screen:

<F2> = System Setup

<F12> = PXE Boot

P/N 7H056 Rev. A00
July 2001

07H056 A00

Authentication?
Keys?
Passwords?

Maintenance

- Address vulnerabilities
- Address theoretical attacks
- Address actual attacks
- Learn from history

And please don’t do this

- **Unpatched** vs. Patched
- **Remote** vs. Local
- **Popular** vs. Obscure
- **Root** vs. User
- **Unauthenticated** vs. Authenticated
- **Reliable** vs. Unreliable
  - **Version independent** not version specific
  - **Bypasses** OS protections

Questions