What's the Worst That Could Happen?

Peyton Engel
Security Engineer
Berbee
peyton.engel@berbee.com

Agenda

- A brief moment of background
- What is "vulnerability?"
  - Do we need to worry whether we have any?
  - What makes a vulnerability severe for us?
- How can we make this less terrifying?
  - A little bit of math
  - A little bit of software
- Inverted security and de-perimeterization
- Interesting and useful observations

- Assumptions:
  - Everyone here runs at least some Windows
  - Everyone has at least one NT domain or Active Directory
- Don't be alarmed if this gets strange: it's all going to turn out OK in the end…

A Few Words About Me Personally

- Dropped out of PhD Program in Russian Literature…
- Two years as "network services coordinator" for Learning Support Services (College of L&S; UW-Madison)
- Dedicated to security engineering since 1998
  - Product deployment (firewalls, etc.)
  - Teaching (application security)
  - Research (DefCon, ToorCon, LISA, software tools, etc.)
  - Assessments, mostly

My Mental Limitations

- I am not a mathematician
- I am not a computer scientist
- It is easy to stump me with questions from either discipline
- If you have one of these two callings, then some of what I say may seem obvious or boring (not XOR!)
- But all is not lost:
  - The things included here are all needed for real-world work
  - Also, it can't possibly be that hard…

Part of Why My Job Is Frustrating (yours too?)

- We hear about "new" vulnerabilities all the time
- Security vendors talk about detection or remediation of vulnerabilities
- On the basis of vulnerabilities identified in various customer systems, I try to convince organizations to make changes
  - Some of the changes are costly
  - Some of the changes are unpleasant
  - Some of the changes are hard to explain
- Spectrum of responses, any of which might represent failure on my part:
  - Nothing – this couldn't possibly be an actual risk
  - Mad panic – OH NOES!!!!!!11!!!! We quit!
- Deciding if/how to address a vulnerability means we need a better understanding what a "vulnerability" really is

What are the Ingredients in a Vulnerability?

1. A problem of some sort must exist
2. The problem must involve a change of security state
3. It must be possible to trigger the problem

- So far, that's a vulnerability… It doesn't mean a great deal to us until:

1. Someone finds out about the vulnerability
2. Someone figures out how to exploit the vulnerability
3. It turns out we own one or more of these "vulnerable" things
4. Someone has the ability to use the exploit on us

- OH NOES!!!1!!!1!!!!oneoneone!!!

Is This a Vulnerability?

```
char infile[80], username[40], mail_file[40],
current_user[40],  tmpstr[40];

/*     snip of some intervening code that doesn't
pertain to this example

       ... blah, blah, blah ... mumble mumble mumble ...

*/

strcpy(current_user, getenv("LOGNAME"));
```

- There's a stack-based buffer overflow (i.e., a flaw)
- This is a SUID binary owned by root (i.e., unauthorized access)
- This is AIX, where environment variables can by 2048 bytes (i.e., it's exploitable)
- OH NOES!!!!11!!1!!!!eleven!!!!

How to be Vulnerability-Free

- Plan 1: Find out about and fix all flaws in all products
  - Not likely; vendors keep releasing patches, indicating that they don't know them all…
  - "Apollo 8 has 5,600,000 parts and one half million systems, subsystems, and assemblies. Even if all functioned with 99.9% reliability, we could still expect 5,600 defects."
    - Jerry Lederer, NASA safety chief (quoted in Collins, Michael. *Carrying the Fire: An Astronaut's Journeys*, New York: Random House, 1974, p. 307)

- Plan 2: Prevent all flaws from being exploitable by anybody
  - Also problematic; generally this would involve denying all access…
  - "The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts."
    - Gene Spafford (quoted in Dewdney, A. K., "Computer Recreations: Of Worms, Viruses and Core War," *Scientific American*, March 1989, p. 110)

"Window of Vulnerability"

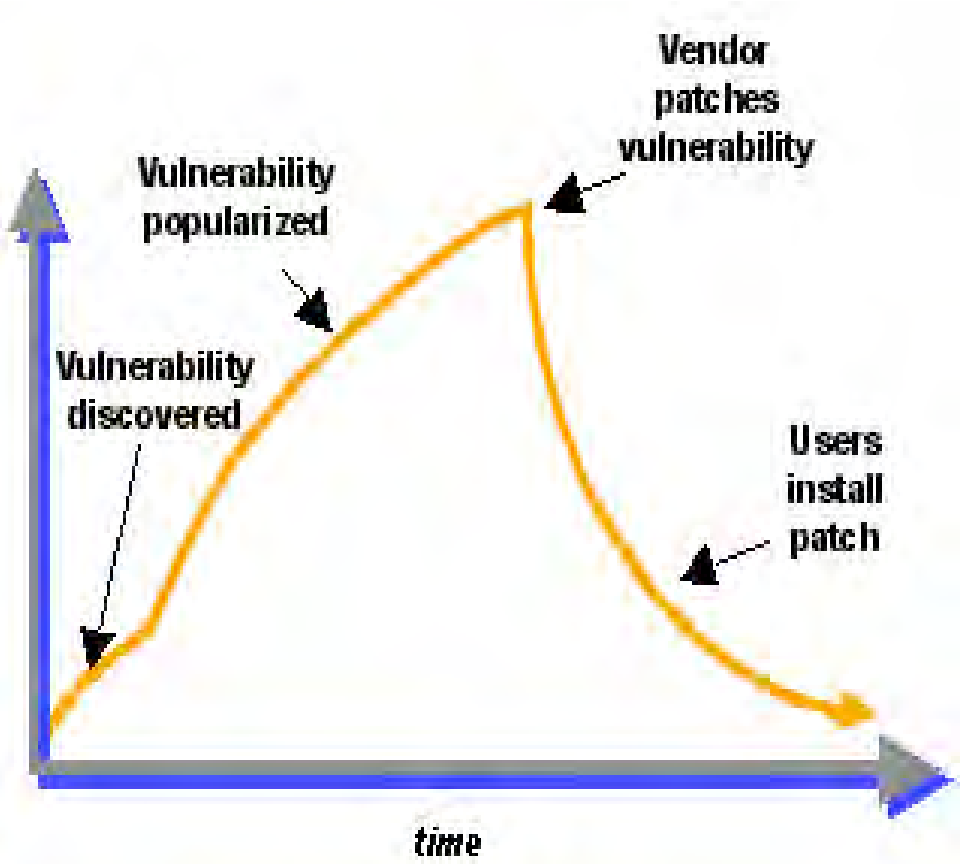


Figure 1

(graphic from counterpane.com)

- Introduced to describe worm/patch cycle
- Note that it is never really 0 (i.e., there are always some vulnerabilities we don't even know about)
- This is the 0-day problem, and we are not likely to solve it any time soon
  - Actually, it has become a new industry…
- Does anybody really think they have 0 vulnerabilities?

Trying to Quantify Risk

- Let's just accept that we have some vulnerabilities, then.

Risk($/year) = SLE($) * ARO(incidents/year)

- So far, we've been thinking in terms of ARO
- We will never be able to limit ARO to 0.
- We will never be able to cap SLE, either, but maybe there is something we can do about it…
  - NB: We are going to use Windows examples, but the principles at stake are reasonably generally applicable

Questions I Want To Answer

1. If a given host on my network is compromised, how severe is the problem?
2. Does having this information help me take any simple steps to make that eventuality less awful?
   - Patching everything ASAP and being a flawless administrator is not simple
   - Neither is deploying a shiny new HIPS everyplace

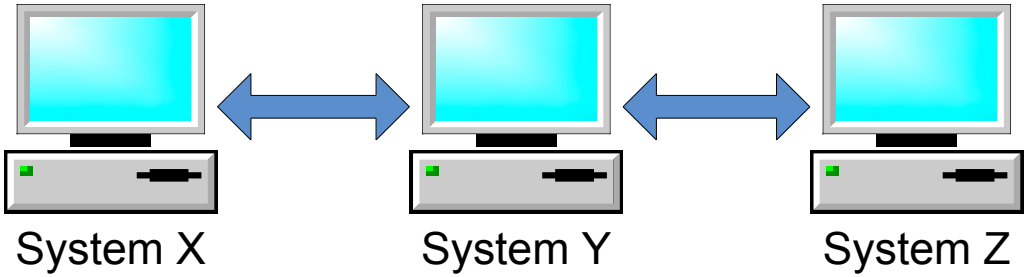
   (these are fine ideas, and worth trying, in many cases, but they're not necessarily easy to do well)
3. Can I make this advice accessible enough to be helpful to people other than security nerds?

(My goals in life: Usefulness, thoroughness, clarity)

Trust Relationships

- A user might have the same username and password on two systems
  - User Peyton on X has the same password as User Peyton on Y
  - User Adam on Y has the same password as User Adam on Z

  (For the time being we are only considering admins)
- We can construct an "Adjacency Matrix" to describe these connections (i.e., if you know all the passwords on a system, you can get to all adjacent systems)



	X	Y	Z
X	0	1	0
Y	1	0	1
Z	0	1	0

What's So Bad About Trust Relationships?

- A single vulnerability might give an attacker access to a great deal of stuff
- Once that happens, it's hard to distinguish between logins by legitimate friends vs. logins by nasty tricksy hobbitses^Wadversaries
- It might be pretty hard, even, to determine if Something Bad™ has happened
- It's might be relatively easy to gather data about various vulnerabilities, but it's hard to spot the relationships that govern how deadly they are
- So we don't know how much to panic! OH NOES!!!@#!!!

Matrix Multliplication

- The adjacency matrix only shows paths of length 1 (i.e. adjacent)
- If we have an adjacency matrix M, then M^2 shows us paths of length 2, and $M + M^2$ shows is paths of length 1 and/or 2…

$$C = \begin{Bmatrix} c_{0,0} & c_{1,0} & \cdots & c_{m,0} \\ c_{0,1} & c_{1,1} & \cdots & c_{m,1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{0,m} & c_{1,m} & \cdots & c_{m,m} \end{Bmatrix}$$

$$c_{i,j} = (a_{i,1} \times b_{1,j}) + (a_{i,2} \times b_{2,j}) + \ldots + (a_{i,m} \times b_{m,j})$$

$$c_{i,j} = \sum_{k=0}^{m} a_{i,k} b_{k,j}$$

	X	Y	Z
X	1 1	1 0	1
Y	1 0	2	1 0
Z	1 1	1 0	1 1

Identity

- Identity: For an operation °, if I is the Identity, then X ° I = X
- The identity element (for scalar multiplication: 1 * x = x)
  - Trivia: name the additive identity

$$I = \begin{Bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{Bmatrix}$$

Inverse

- Inverse: For an operation °, if I is the Identity, and X^{-1} is the inverse of X, then $X \circ X^{-1} = I$
- 1/X is the multiplicative inverse of X
  - Trivia: what's additive inverse?

$$I = \begin{Bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{Bmatrix}$$

$$x \times x^{-1} = \frac{x}{x} = 1; MM^{-1} = I$$

Geometric Series

- What if it were possible to calculate a single matrix that showed the existence of paths of arbitrary length?
- Consider this polynomial
- Its analog for matrix math ("Transitive Closure")

$$(1-x)^{-1} = \frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots$$

$$(I-M)^{-1} = I + M + M^2 + M^3 + \dots$$

A Few Review Words About Windows Crypto

- LANMAN hashses
  - Passwords divided up into 7-character blocks (i.e., never a need to crack anything > 7 char)
  - Passwords uppercased (i.e., no need to search lowercase keyspace)
- Both LANMAN and NTLM
  - No salt
  - Hashes can be replayed (i.e., no need to crack encrypted passwords)
- Cached credentials
  - MD4(NTLM Hash + Username)
  - Username is the salt; can be cracked, but takes a little while

Our Initial Process…

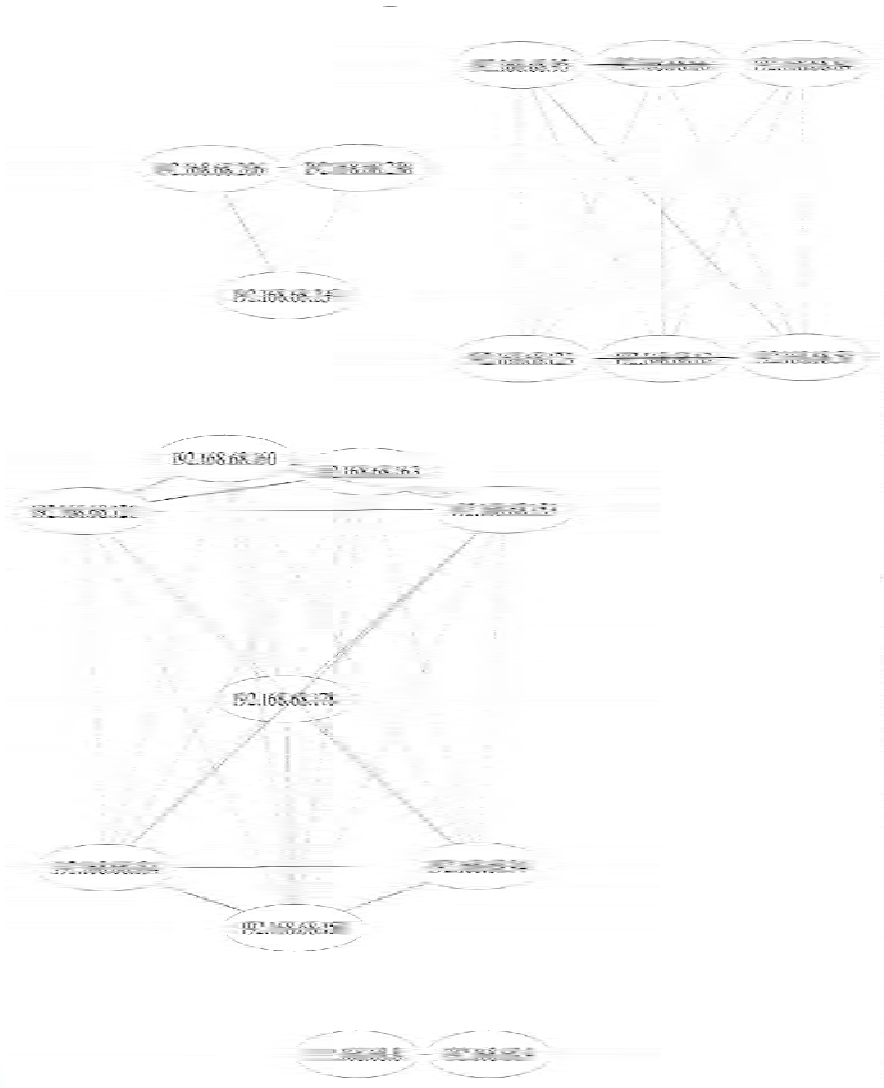
- Log in as an administrator to all machines in your network (all this is easily scripted)
- Pwdump > my_IP_address.pwdump

```
Administrator:500:3F4954CC24F78E1AAAD3B435B51404EE:23
0CDBB756D4DC5B6E4AC543BE6FCEF4:::
```

- For each password/hash combo, if it exists on another machine, those two are “adjacent”

Now we can draw connectivity graphs!

- If system X is hacked, what other systems should be considered indirectly hacked?
  - Financial institution (120 or so hosts)
- It would be nice to play what-if games, also: "what if we eliminate the help desk account?"



So, we're part-way there

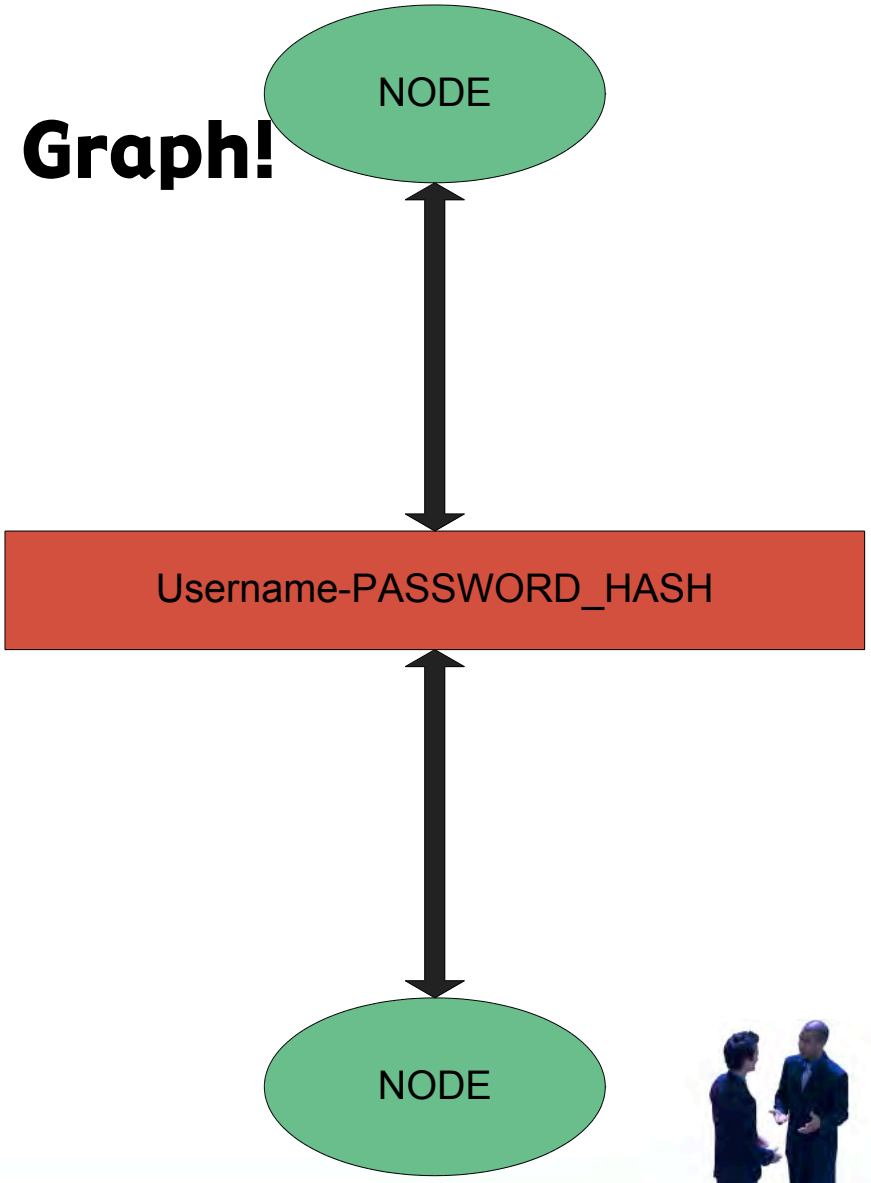
- We can now see the problems with any one system being hacked
- But we don't know what to do about it…
- Plus, we don't have easy access to information about local accounts…
- And we haven't got cached credentials yet…
- Also, our anti-virus software causes systems to commit suicide when we run pwdump…

Some tools to make it easier

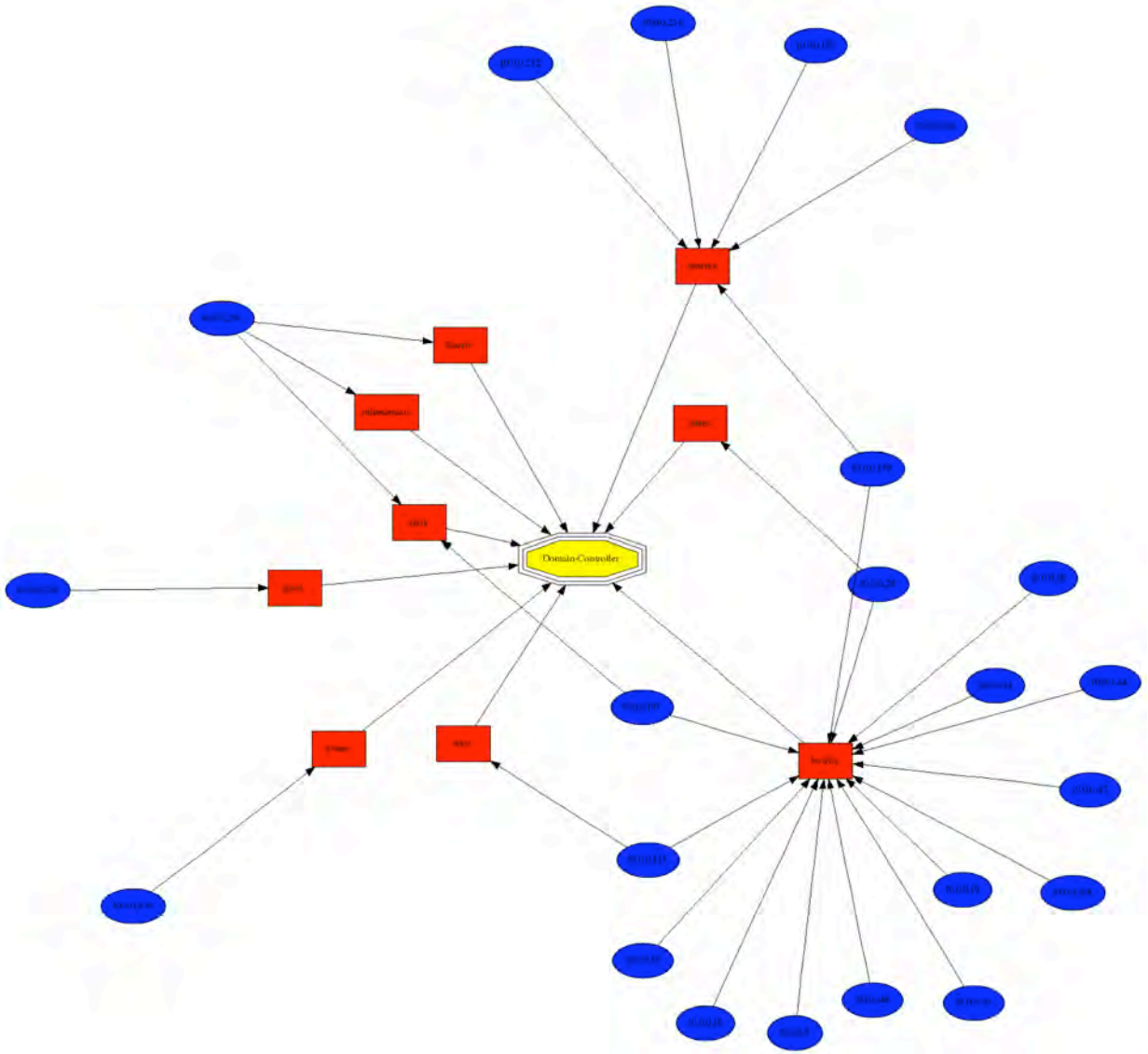
- OWNR : A modular NetBIOS information-gathering tool
  - Released at DefCon 13; simple Win32 API wrappers
  - Designed to give output in a format frendly to grep|cut
- fgdump : Replacement for pwdump and cachedump
  - Released at ToorCon 7; still might crash some stuff
  - Smart enough to disable some antivirus, though
  - Opportunistic about writeable shares
- pwdumpToMatrix.pl : parse hordes of password files, and describe adjacencies
  - Can whitelist or blacklist accounts
  - Output is just text files (matrix and character-separated)

Plus, A New Kind of Graph!

- E-R style!
- Show systems, and the usernames/passwords that connect them
- Like a subway map
- Let's look at a couple…



Example Graphs

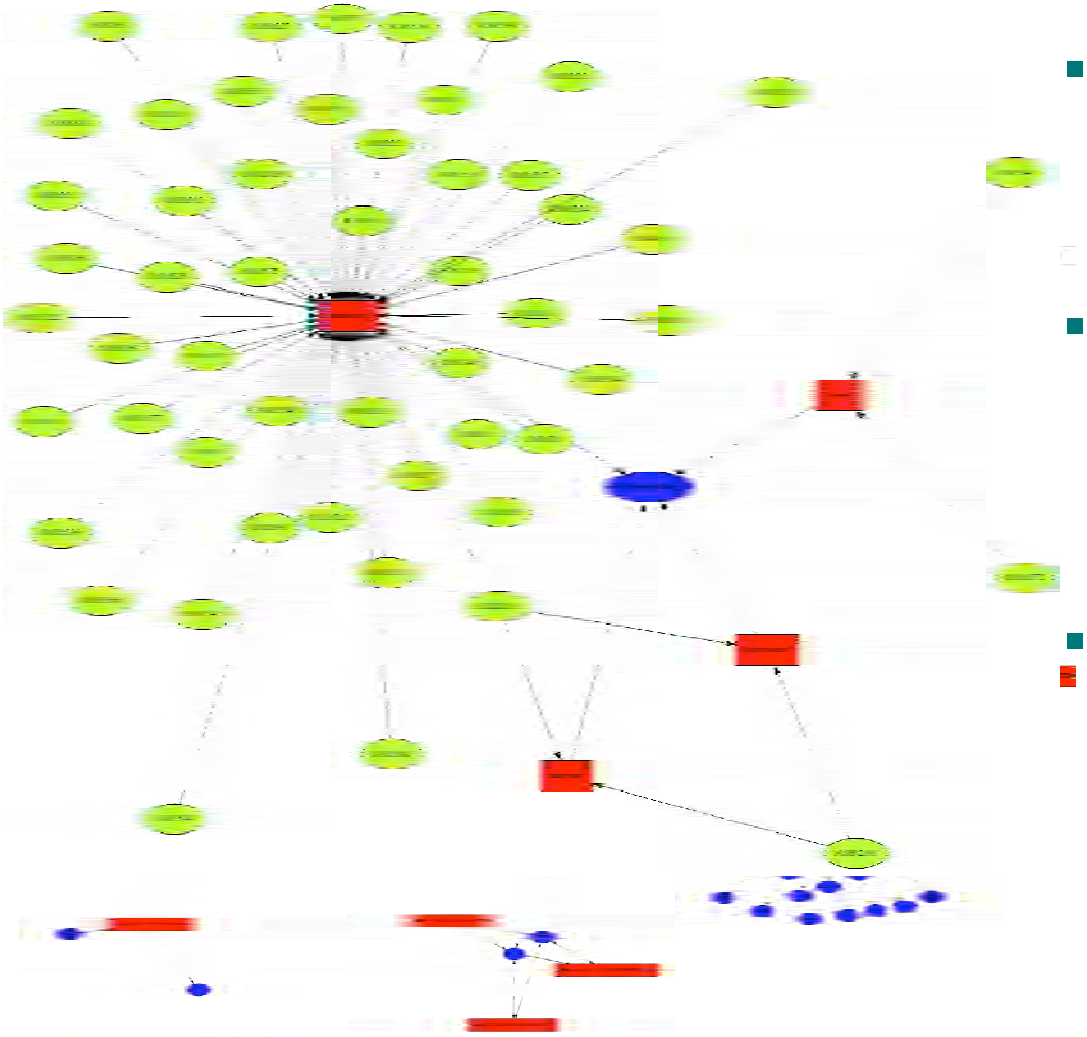


- School District: 314 systems; 312 interconnected
- School District: Administrator is disabled
- Insurance Agency: doing better
- Insurance Agency: cached credentials

The Actual Process Used To Build These

- Gather information from a network fgdump & OWNR, wrapped in Perl
- Prune out nonsense (DCs, IUSR_, ...)
- Process that raw data into a format that describes relationships (pwdumpToMatrix.pl)
- Convert the descriptions of relationships to descriptions of graph nodes and edges
- Cleanup (highlight DCs, special cases)
- Render those descriptions as an image (GraphViz)

More Example Graphs



- Small manufacturing company; two problematic accounts
- Bigger manufacturing company
  - interesting clusters
  - Note the DMZ hosts
- Same manufacturer; cached credentials

Making some inferences from graphs

- There are things you can do to lessen the impact of a new vulnerability
  - Principle of "Least Privilege"
  - Reduce sharing of local accounts
  - Turn off cached credentials where not needed (and setting number of cached credentials to 1)
- We can begin to think in terms of vulnerabilities we don't yet know about.

The Moral of This Little Story

- We're all always vulnerable to something (or at least we should assume that's the case)
- Nevertheless, there are things we can do to lessen the impact of any vulnerabilties that are discovered
- Sometimes, with some relatively simple changes, we can both
  - Reduce our attack surface
  - Give ourselves better hope at containing an incident

What Was With All That Math?

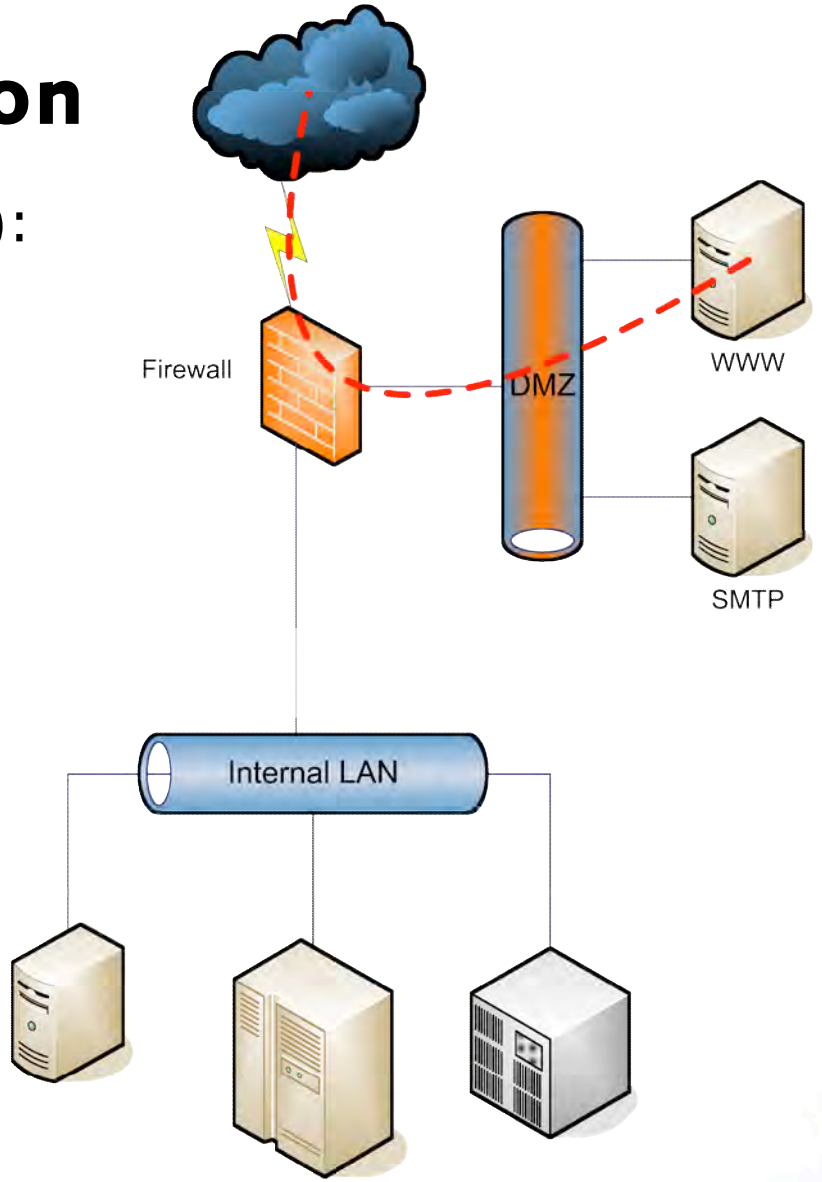
- Some graphs might be hard to visualize in their entirety
- We want to be able to answer other questions:
  - What's the most commonly shared user account?
  - On what computer's security do the most others depend
  - What's the list of systems accessible from host X?
  - Can we compare the "density" of networks?

Inverted Security

- Forrester Report: “Let’s get rid of firewalls”
  - Patching systems
  - Disable unnecessary services
  - Good administration
- Some sense to it, though
  - Put security on the thing to be secured
- Pertinent examples
  - 802.1x access controls on network jacks
  - Host-based intrusion prevention for workstations

De-perimeterization

- Previous LISA talk (2003): dissolution of network boundaries
  - Porous firewalls
  - Mingling of layer 2 and layer 3 separation
  - Extruded networks
  - Wireless networks
  - Apps with newly discovered functionality
- New ingredient: device convergence
- All this amounts to de facto de-perimeterization



De-Perimeterization as a Goal

- The Jericho Group
  - http://www.opengroup.org/jericho
- "open standards to enable secure and boundaryless information flows across organizations"
- "A new approach is needed, to move from the traditional network perimeter down to the individual networked computers and devices – and ultimately to the level of the data being sent over the networks."

But what is a network perimeter, really?

- An attempt to enforce a belief about where "outside" should begin and "inside" should end
  - It's often not where we think it is
- The hacker perspective: "hard crunchy shell with a soft chewy center"
- If we place security on our hosts, but have trust relationships between them, ***we have created a new perimeter***

Concluding Thoughts

- We often find ourselves making dangerous assumptions
  - In our network designs (i.e., failure to account for endpoints)
  - In our organizations (dividing up security responsibilities artificially)
  - In the way we look at the systems we build
- Why are network perimeters such a big deal?
  - They represent assumptions about where defenses should be placed
  - "Hard crunchy outside" with "soft chewy center" :
    - single points of failure in defenses
    - large-scale consequences
- Current buzzword: "de-perimeterization" (inverted security + "access anything from anywhere, securely")
- As we do this, though, we're going to be finding new and dangerous perimeters.

Trying this at home

- Afterglow: Converting character-delimited descriptions of connections into descriptions of nodes and edges ("dot" language):
  - http://afterglow.sourceforge.net/
- GraphViz: Converting dot files into images:
  - http://www.graphviz.org/
- Doing matrix algebra: the J programming language
  - http://www.jsoftware.com
- Gathering user account and password information from systems:
  - OWNR, fgdump, pwdumpToMatrix (http://www.foofus.net)
  - (or pwdump, cachedump)
  - perl

Bibliography

- Burgess, M., Canright, G., Hassel Stang, T., Pourbayat, F., Engø, K., Weltzie, Å., "Archipelago: A Network Security Analysis Tool," ***Proceedings of the Seventeenth Systems Administration Conference (LISA XVII): 153***, Berkeley: USENIX Association, 2003.
- Carré, Bernard. ***Graphs and Networks***, Oxford: Clarendon Press, 1979.
- Corman, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. ***Introduction to Algorithms***, Cambridge: MIT Press, 1990.
- Halprin, Geoff. ***A System Administrator's Guide to Auditing***, Berkeley: USENIX Association, 2000.

Thanks! Questions?