Secure Collaboration

External Links

- Sometimes, there are business needs for links to outside parties: suppliers, customers, joint venture partners, etc.
- There are more of these than you think—a large company will likely have thousands of them
- The technical people rarely get to say “no”

Many Types!

- Simple request/response
- Simple transactions
- Multiple databases
- Complex interactions

Terms

- Technical folks rarely get to set the terms
- That’s a business decision, made at the business layer
- If the risk is too high, you have to propose alternatives (more on this later)
- But details matter

Request/Reponse to an Open API

- Buying supplies: relatively easy; call out through your firewall to the supplier's open web site
- Security issues are more business-like: who can place orders, how does internal chargeback work, etc.
- Selling to another business: similar to the secure e-commerce scenario—if you can't do that, you have bigger problems. . .

Sharing Databases

- Suppose you need to share access to a database
- Do you
  1. Punch a hole in the firewall?
  2. Set up an application-layer proxy?
  3. Something else?
- It depends. . .

Punch a Hole

- Can work, for comparatively simple database needs
- Use database logins and permissions to restrict access
- For example: give the remote party access to some columns but not others, or perhaps read-only access to some and read-write access to others
- This requires care when changes are made to the database, to ensure that the proper permissions are set for new or changed elements
- Also: can you log remote activity properly?

Application Proxy

- What is the language your proxy accepts?
- Full SQL? Are you sure you're parsing it correctly?
- Your own language? Have you specified it and implemented it correctly?

There's a Tradeoff

- Languages and implementations are hard to get right
- So are permission settings
- Which is easier?
- The crucial factor: complexity. Complex tables and structures suggest that code is better; a complex language suggests that permissions are better
- Also: which do you think is more likely to become more complex? Do you anticipate adding columns or tables?

The Business Approach

- Businesses trust other businesses for lots of things
- Payment for goods or services delivered is often “net 30”—pay within 30 days
- It can be a matter of trust and experience:

  A small business may use shorter payment terms, like net 10, with new customers or customers that tend to pay late. Once the customer starts paying on time, the business may extend longer payment terms like net 30 or net 60.

- Should the same apply to interconnections?

Russian Saying: Trust But Verify

- *All* interconnections should be done by a formal agreement
- The agreement should specify—in human-readable terms—what is and is not allowed
- The software should log all requests; these logs should be analyzed
- You generally want to take reasonable technical precautions as well; if nothing else, the other company might itself have been hacked
- Target and Home Depot were both hacked via outside parties who had authorized access to their networks

Network Segmentation

- Sometimes, an outside party need broad access for a specialized function
- Example: Target used an outside party to manage the HVAC (heating, ventilation, and air conditioning) systems in its stores
- Firewall off that network (or networks) from the rest of the corporate net, and give the outside party access to that special network
- Of course, if you get that wrong—well, that's how Target was hacked...

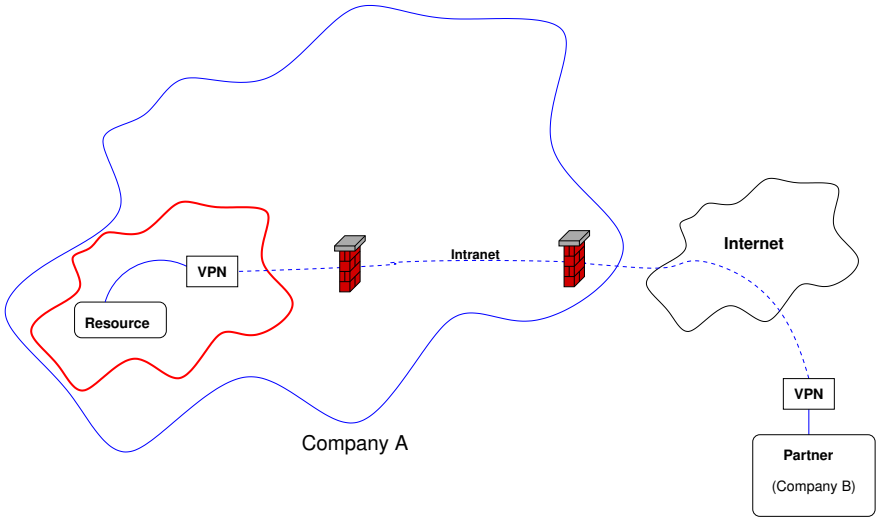
(Network Segmentation)

- Network segmentation can be a good idea even without outsiders
- Example: with Home Depot and Target, the attackers planted card number skimmers in the firmware for the Point of Sale credit card readers—why were they accessible?
- Use VLANs, network ACLs, and more, to isolate sensitive systems

Shared Enclaves

- Network segmentation is a special case—we sometimes need a whole separate area for joint work
- Firewall it off from the rest of your company, too

Shared Enclave



Using a Shared Enclave

- The shared enclave lives inside on party's network
- It's firewalled off from that network—no *outbound* calls to that intranet are permitted
- The other party uses a VPN to traverse not just the Internet but also the intranet until it is inside the enclave's firewall

Calling In

- How do Company A's employees connect to the shared enclave?
- More precisely, what are the security policies and rules?
- If policy permits general connection, just have rules on that firewall permitting the connections
- Or: have some sort of access control
- A VPN within Company A, to connect to the VPN server inside the shared enclave?

Calling In from Company B

- Simplest way: give each authorized employee VPN credentials
- But how is that managed on Company A's VPN server?
- That is, who adds and deletes logins?

Managing Access

- It is unreasonable for Company A to decide who in Company B is part of their team
- If A controls the VPN, but B sets its login list, you need either shared administration—risky!—or secure communications from Company B's administrator to Company A's
- Possible answer: dual enclaves, one at either end, with a VPN between them

Dual Enclaves

- VPNs are networks—they carry traffic both ways
- What stops unscrupulous Company A employees from using the VPN to get into Company B?
- It may need its own enclave, to protect it from Company A
- This also solves the access management problem—they control their employees' access to their enclave
- Bonus: some resources can be there, too, acessible to both parties

Analyzing the Shared Enclave

- Where did this architecture come from?
- What is the role of each component?
- Why do we think it’s secure?

Security Analysis

- Company A wishes to protect its other computers
- The shared machines are either susceptible to penetration or need to permit sufficient access to Company B that there is automatic access from them for B's employees to reach the rest of A
- We therefore need to protect Company A from the shared machines—hence the enclave and the firewall

A Look Back

- Let’s look back at the shared database scenario
- That is a machine that can (in theory) protect itself
- In other words, no firewall is needed to protect Company A from it
- We may still have the problem of managing credentials for employees of Company B

Encryption?

- Should the VPN be encrypted?
- Should it? Of course—VPNs are always encrypted!
- Or are they?

Types of VPNs

- Some ISPs will provide you with a VPN—a way to connect two or more locations via a network that's not accessible to other customers
- There are many, many ways to do this sort of separation; cryptography is one but not the only one
- Example: a different wavelength ("lambda") on a fiber
- Example: network technologies such as ATM or frame relay
- Example: a subchannel of a faster link—commonly done via *add-drop multiplexers*
- Example: IP-in-IP
- Many more

Security of Unencrypted VPNs

- If you leased a circuit or fiber from the phone company to connect two sites, would you encrypt it?
- Do you encrypt your landline calls?
- Generally, the answer is no—so why should VPNs be different?

However. . .

- What is your threat model?
- Are intelligence agencies after you?
- Are some links exceptionally susceptible to interception?
- Are you worried about traffic within Company B before the VPN gateway?
- If IP-in-IP is used, how do you authenticate the far end if you don’t use cryptography?

Managing Interconnections

- Whether it's a simple database, several databases, or a shared enclave, you have to punch holes in the firewall
- You'll need such holes for every external connection
- The larger the company, the more there will be—you have to manage them

Managing Interconnections

- You need an approval process
- You need agreement on approval criteria—that may be a CISO function
- You need to track the firewall rules and network configuration changes for every interconnection
- You won't always be told when some project shuts down, so you have to be proactive
- This takes process and infrastructure

Policy versus Mechanism

- I can’t tell you what your policies should be
- They’re generally negotiated, often at a high level—but make sure that security and sysadmin voices are heard
- However, I can describe the (generic) mechanisms that are usually necessary

Requesting an Interconnection

- *All* requests must go through the formal process
- A good mechanism: an *issue tracker*
  - Create a ticket via email or web; the requester receives a ticket number that can be used to check the status
  - Requests must always list a couple of levels of management, in case of employee turnover
  - Requests must have an expiration date: how long will this firewall hole be needed?
  - The tracker can show different states: awaiting approval, approved, active, rejected, retired, etc.
  - It also helps the firewall admins keep track of what needs to be done

Criteria

- The precise criteria use are, of course, company-specific
- They may be different for different parts of the company–Research may have looser rules than, say, HR or Legal
- However: you always need a *business* (or other mission) justification

Setting Firewall Rules

- *Never* simply change your firewall rules
- Instead, use a version control system, e.g., `git`, to keep track of changes
- *Always* list the ticket number in the commit message—that links particular rules to particular requests
- Better yet, use database-driven configuration (see the reading)—but again, link each firewall hole to a ticket number
- This makes it easy to back out rules when they're no longer needed
- Query: how do you handle a rule change that is needed for more than one ticket?

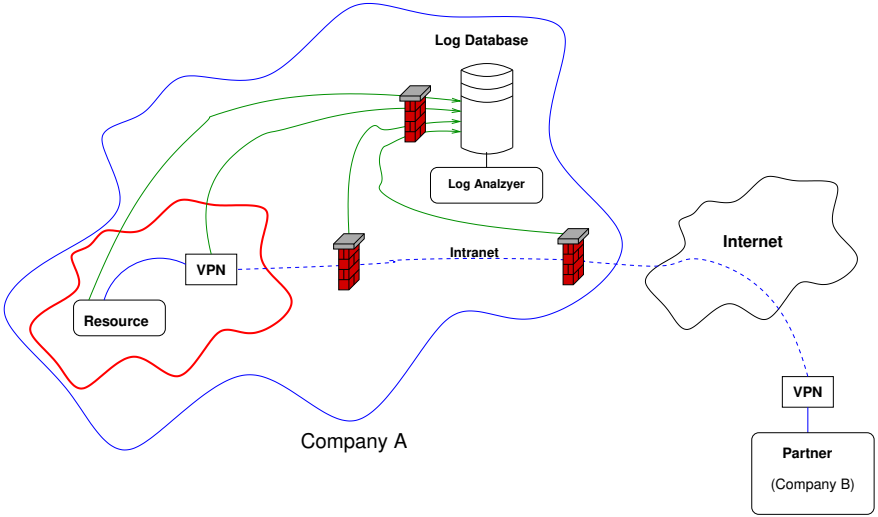
Expiration

- Make sure that rule changes expire
- Notify the requester before the expiration; see if they still need it
- Delete on no response? Probably not; try to contact a manager. Also, look at traffic logs to see if that rule is still passing traffic
- (But if you turn off a rule that is in use, you *will* get a response. . .)
- Precise policies here are very site-specific

Logging

- I said that all requests should be logged
- Doing that properly is vital—and not obvious

Logging


Log Database
Log Analzyer
Intranet
VPN
Resource
Company A
Internet
VPN
Partner
(Company B)

Logging Design

- *Every* element should create log file entries
- They should be written to a database
- This database *must* be protected:

  > *It is often a good idea to keep your files in an off-machine logging drop safe. Hackers generally go after the log files before they do anything else, even before they plant their back doors and Trojan horses. You're much more likely to detect any successful intrusions if the log files are on the protected inside machine.*
  >
  > —Cheswick and Bellovin, 1994

- And: be sure to analyze your logs
- N.B.: all of this applies even if you aren't dealing with interconnections to other companies

Why a Database?

- You're getting data from many different sources
- Many analyses require correlating log entries from various places
- Once, you could do this by hand; today, there are too many devices generating too much data

Log Analysis

- If you never look at your log entries, you may as well not have any
- Always do intrusion detection on your log databases: look for specific evidence, and look for anomalies
- Anomaly detection always needs tuning—but you'll never find attackers with just known patterns

After an Attack

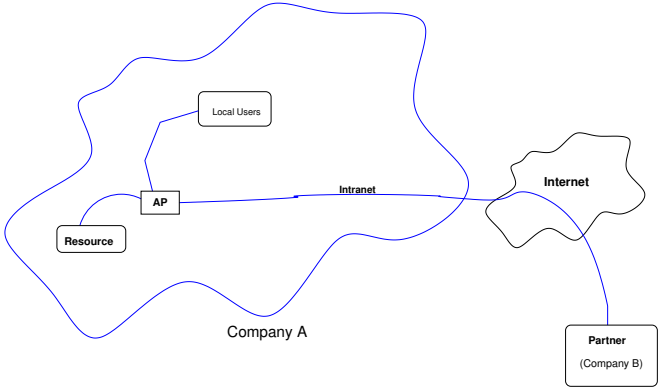
- Your log database will show you what happened
- This is how you can tell how the attackers got in—and perhaps what they did
- Some of this information might be legal evidence—more on that later in the term

What About a Zero Trust Architeture?

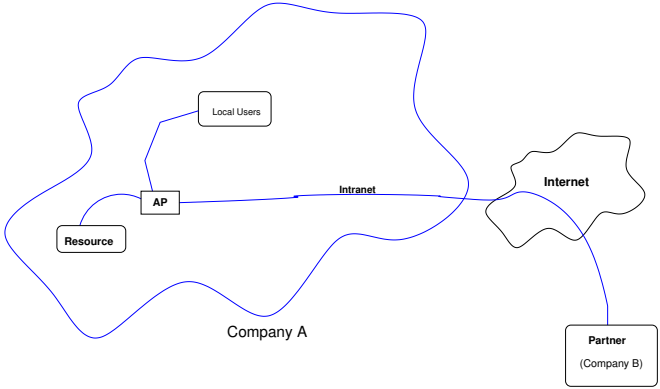
Issues

- Suppose we want a zero trust version
- The previous architecture relies on firewalls and VPNs
- But zero trust architectures don’t use those
- What should we change? What do we need to add?

A First Attempt (Logging Deliberately Omitted)



A First Attempt (Logging Deliberately Omitted)

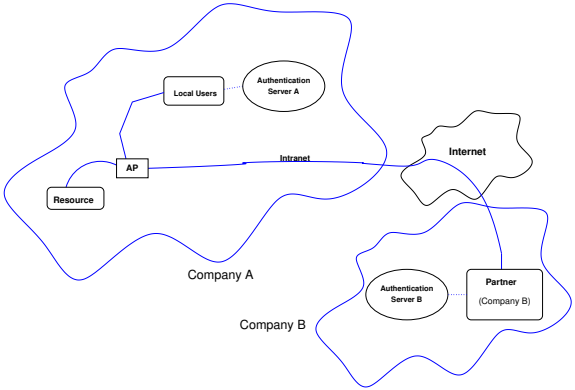


What's wrong?

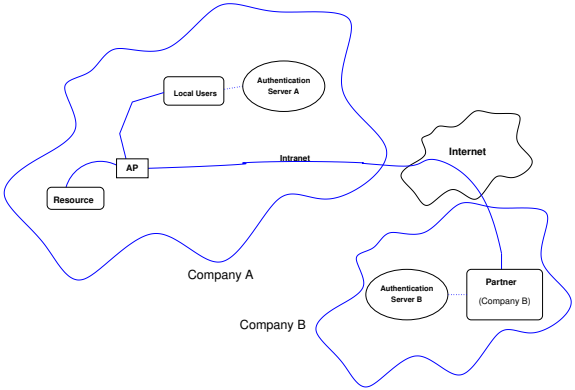
Authentication!

- Company A can authenticate its users, but can’t authenticate Company B’s
- We have to leave it to Company B to authenticate its own users

Second Attempt



Second Attempt

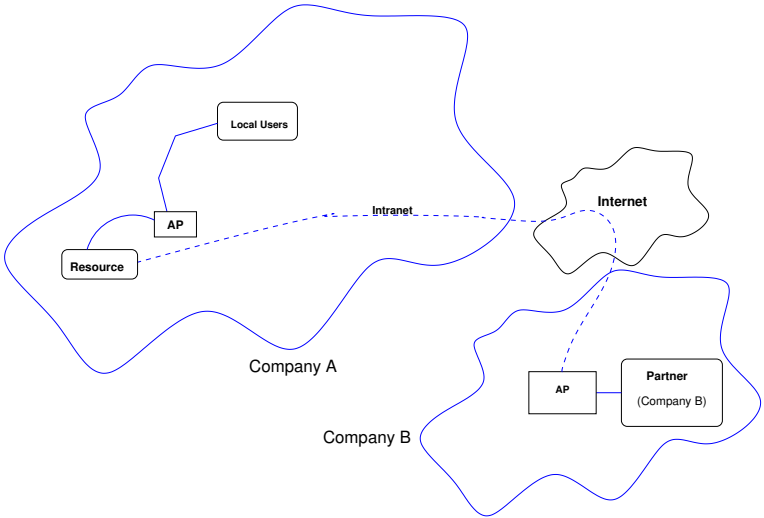


What's wrong now?

Authorization

- Most likely, not all employees of Company B are *authorized* to work on the joint project
- The AP can't know who is—that's Company B's decision
- We have to leave that in their hands

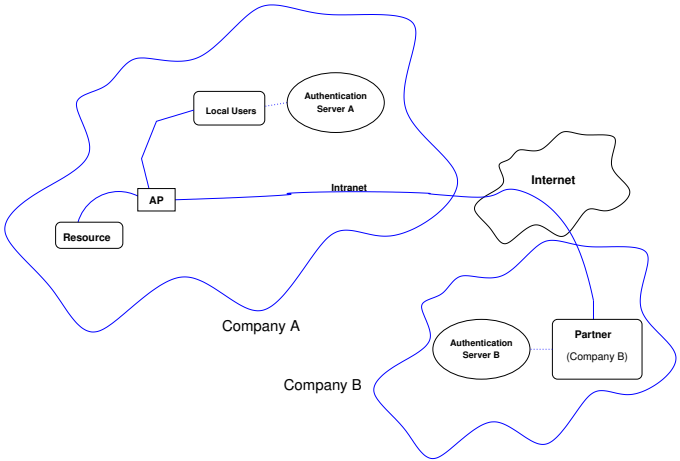
Third Attempt



A Second Access Proxy

- Company B runs its own Access Proxy
- All authorization for Company B employees is done there
- There is a secure tunnel from their AP to the shared resource

An Alternative Design



- Use just one AP
- Have the APs securely include authorization attributes in the authentication “ticket”
- Company A’s AP can validate it

Logging in a Zero Trust Environment

- We still need logging
- In the dual AP variant, we need Company B's AP to send log messages to Company A's log server
- In the single AP variant, probably no change is needed, though it might be useful for Company B's authentication server to note attempts to spoof authorized users of the shared resource

Security Analysis

- The entire premise is that Company A uses zero trust—there's no incremental risk from the concept of multiple access
- There is heavy reliance on Company B's authentication and authorization procedures and code
- The dual AP model might be slightly more secure, since it simplifies administration: the administrator of the corporate authentication server doesn't have to know about every joint venture
- It doesn't matter if Company B uses zero trust or not, though if they don't they may need an extra hole in their firewall for the tunnel from their AP

Collaboration

Collaboration

- Collaboration is hard but necessary
- It's hard because by definition, it involves drilling holes in your security perimeter
- It's necessary because that's generally the best way to run a business

Questions?

(A group of northern shoveler ducks, collectively stirring up food from the bottom, Central Park, January 18, 2021)