Real World PeopleSoft Vulnerabilities
A "Most Wanted" List

Ben Klang

System Efficiency

Introduction**:** Is your ERP system secure? How do you know? Identify and resolve the top 4 security Mis-configurations that can lead to unauthorized access of your system as well as ensuring only the necessary access is needed:

- Default usernames and passwords (WebLogic, Tuxedo, OS, PeopleSoft, Oracle)
- Weak node (SSO) authentication (cookie forgery and inter-phase bad trust)
- The information you give away and don't even know it (information disclosure in cookies, logfiles, error messages)
- Too much privilege: not everyone needs to have all permissions

Security is a practice similar to medicine: Visit 5 different specialists and get five different answers. Determining which answer is best can be an exercise in futility. The good news is that, as in health, taking basic precautions can go a long way toward avoiding a visit from to the Doctor in the first place. This paper addresses some of the most commonly discovered yet easily preventable problems within PeopleSoft deployments. Each and every topic covered here has caught "in the wild" and, unfortunately, not just once. Take this free advice to help your organization before you consider helping your security vendors make their quotas.

Each year the Computer Security Institute, formerly in association with the Federal Bureau of Investigation, sends out a security survey to a group of its members. The response to that survey is published as the CSI Computer Crime and Security Survey report. Now in its thirteenth year, it provides an excellent and reliable benchmark for trends in security incidents. 2007 was a remarkable year. Reversing a five year downward trend, the Average Annual Loss reported increased by nearly double to $350,424. Of those organizations reporting an incident, nearly one in five claimed to have been specifically targeted. Perhaps most interesting and prescient is the finding that the previously undisputed king of incidents, the Virus, has been usurped by Financial Fraud. The combined weight of this should speak loudly to PeopleSoft administrators to take a look at deployed systems and make sure the systems do not contribute to these statistics.

System Efficiency's Most Wanted

1. **Default User Accounts**
2. **Weak Node Passwords**
3. **"Most Privileged" Security Model**
4. **Shared Accounts**
5. **Process Blindness**
6. **Policy Amnesia**

Default User Accounts

Many applications ship with default usernames and passwords. This practice, though not ideal, is common in the industry. The challenge unique to PeopleSoft is that when you deploy PeopleSoft, you do not deploy a single application. Instead perhaps 20 or more applications, each potentially with its own usernames and passwords, are deployed in support of the megalithic PeopleSoft. Oracle, WebLogic or WebSphere, Tuxedo, PeopleTools, not to mention the application itself, to name a few. Each comes with its own defaults and each must be located and secured. Some common defaults include:

- Oracle
  - Username: “internal” Password: “oracle”
  - Username: “system” Password: “manager”
  - Username “sys” Password “change_on_install”
- BEA WebLogic
  - Username: “weblogic” Password: “weblogic”
  - Username: “system” Password: “weblogic”
- BEA Tuxedo
  - Username: “admin” Password: “password”
  - Username: “tpsysadm” Password: “password”
  - Username: “tpsysoper” Password: “password”

Weak Node Passwords

Another often overlooked area of concern is the PeopleSoft inter-node authentication system. PeopleTools allows bypassing the sign-in process under certain trusted circumstances. This trust is managed in the system by assigning either a node password or a node certificate. The potential for vulnerability occurs when a node password is either too weak (therefore vulnerable to brute force) or is reused between systems. The most commonly seen and easily exploited condition occurs when the node password is shared between development and production environments. In this case once a user logs into a development environment simply changing the URL to the production system will cause the pre-configured trust to allow the user directly into the production environment without being prompted for a username or password.

Shared Accounts

Since the dawn of multi-user systems the importance of keeping track of individual users has been proven again and again. Yet human nature leads many organizations astray, yielding to the temptation to allow multiple users to share a single account. This practice has many drawbacks. Perhaps the most serious of these is the inability to audit. With the sensitivity of financial and human resource systems the inability to audit is simply unacceptable. What happens when vacation days go unaccounted? When benefits are mysteriously changed? Who did that? Even as many of those changes are not made in bad faith, the inability to track these changes can have dire consequences. Another serious limitation of shared accounts is user security. Should a password ever need to be changed, there is now a management problem caused by the requirement to communicate this changed single password to many users. It may even lead to resistance in changing that password, potentially allowing a discharged employee access to a system to which they no longer should have access. While this may seem obvious, or even silly, it is a real problem that occurs every day in organizations. An all-too-common example of this includes many users sharing the “root” password (or “Administrator” on Windows).

“Most Privilege” Security Model

There is a practice in the security community known as “Least Privilege.” This practice states that each user within a system should only be given the privileges absolutely required to complete their assigned tasks. By assigning security in this restrictive way significant potential for abuse is eliminated. Unfortunately, again due to human nature, many organizations practice the opposite model: Most Privilege. It is often easier for an administrator to grant wide permissions rather than identify the specific requirements for each user. Not only does this raise the potential for abuse, the potential for errors skyrockets as well. Small errors which might otherwise be contained to a single process may now impact an entire system or department. Common examples of the “Most Privilege” Security Model include such go-to actions as “chmod 777” (grant all users all permissions on a given file or directory) or the practice of granting Administrator roles to all users. A shared “root” or “Administrator” password can also fall into this category as this has been seen frequently in situations

where users get frustrated with the inability to complete some assigned task and an overworked system administrator team simply shares this access rather than finding the right combination of settings to achieve the goals. Despite the best expressed intentions of "coming back later to clean up," rarely, if ever, does this occur leaving an organization vulnerable.

Process Blindness

Process Blindness occurs when departments or groups whose tasks interleave or overlap fail to properly communicate with each other. The most extreme examples occur when a security department within an organization does not properly communicate with its constituent groups. This often leads to security mandates which have no foundation within reality. Ineffective security policy breeds two very significant problems: First, users are not automatons. They can sense this mismatch between the security policies and the "real world" and they will find ways to avoid, ignore or even circumvent security controls that are in place. Second, it may create a false sense of comfort to management that "we are covered." This false sense of security can put blinders to real security risks and retard proper reactions to an actual incident. With these blinders in place the scope of a security problem becomes magnified exponentially. For these reasons there is an adage that "Bad security is worse than no security."

Policy Amnesia

One important task for any system administrator, security administrator or manager is to ensure that once good security policies have been defined, they are enforced. Unfortunately, for a variety of reasons (perhaps including Process Blindness), users have a tendency to be unaware of the security policies. Perhaps this is due to a lack of initial training. Maybe it stems from changes to job functions and roles or maybe they simply forgot over time. The result is the same: policy becomes ineffective as users fail to follow the rules. It is imperative that any organization follow up on the policy with training, audits and enforcement programs. These do not need to be heavy-handed or omnipresent. Perhaps a simple monthly email with a "policy of the day" or an hour each week talking to a sample handful of users ensuring the policies are understood and followed. At all costs, an organization should avoid writing a thick policy encyclopedia and then promptly shelving the document never to be seen again.

Conclusion

While it may seem a daunting task, PeopleSoft security can be greatly and tangibly improved by taking some common-sense approaches. Whether your organization has forgotten its own policies (or never defined them) or today shares the "root" password to all of your servers, simple, practical steps can be taken. These common security issues, each seen and addressed by System Efficiency consultants while on engagement, are often the equivalent of "securing the door and opening the window." By searching for these "windows" of Weak Node Password and Default User Accounts into your organization, you can prevent many possible incidents before they occur.